<?xml version='1.0' encoding='utf-8'?><ResultsSession buildId="Default Project-2023-03-20" date="03/20/23 11:40:42" machine="DESKTOP-DCD46AU" project="Default Project" tag="cpptest\_settings" time="2023-03-20T11:40:42+08:00" toolName="C++test" toolVer="2022.1.0.20220615B1827" user="devtest">

<TestConfig machine="DESKTOP-DCD46AU" name="cpptest\_settings" pseudoUrl="c++test.file:/c:/jenkins/workspace/cicd.findings.cpptest.professional.static.analysis.report/conf/pro/cpptest\_settings.properties" user="devtest"/>

<Authors>

<Author id="dev1" name="devtest"/>

</Authors>

<VersionInfos>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.execution" resultId="IExecutionViolation" ver="2"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.execution" resultId="InsureViolation" ver="3"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.scope" resultId="IScopeProjectInfo" ver="3"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.standards" resultId="IMetricsViolation" ver="2"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.standards" resultId="IDupCodeViolation" ver="2"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.standards" resultId="IMetricEstimationInfo" ver="2"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.standards" resultId="IFlowAnalysisViolation" ver="2"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.standards" resultId="ICodingStandardsViolation" ver="2"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.generation" resultId="IGenerationInfo" ver="2"/>

<StorageInfo ownerId="com.parasoft.xtest.checkers.api.generation" resultId="IGenerationProjectInfo" ver="2"/>

</VersionInfos>

<Scope>

<ProjectInformations>

<ScopeProjectInfo fltFiles="6" fltLns="245" project="/cicd.findings.cpptest.professional.static.analysis.report" totFiles="6" totLns="245"/>

</ProjectInformations>

</Scope>

<CodingStandards ownerId="com.parasoft.xtest.checkers.api.standards" time="0:01:19">

<Goal mode="1" name="Static"/>

<Projects>

<Project bdCheckedFiles="4" bdTotalFiles="4" checkedFiles="6" checkedLns="245" name="cicd.findings.cpptest.professional.static.analysis.report" qfixErrs="0" suppErrs="0" totErrs="3236" totFiles="6" totLns="245"/>

</Projects>

<Rules>

<RulesList>

<Rule cat="APSC\_DV-000160" desc="Do not use weak encryption functions" id="APSC\_DV-000160-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-000170" desc="Do not use weak encryption functions" id="APSC\_DV-000170-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-000480" desc="Protect against SQL injection" id="APSC\_DV-000480-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-000500" desc="Observe correct revocation order while relinquishing privileges" id="APSC\_DV-000500-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-000650" desc="Do not print potentially sensitive information, resulting from an application error into exception messages" id="APSC\_DV-000650-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001290" desc="Protect against SQL injection" id="APSC\_DV-001290-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001290" desc="Untrusted data is used as a loop boundary" id="APSC\_DV-001290-b" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="APSC\_DV-001290" desc="Avoid passing user input into methods as parameters" id="APSC\_DV-001290-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001290" desc="Avoid using unsecured shell functions that may be affected by shell metacharacters" id="APSC\_DV-001290-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001300" desc="Protect against SQL injection" id="APSC\_DV-001300-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001740" desc="Avoid passing sensitive data to functions that write to log files" id="APSC\_DV-001740-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001750" desc="Avoid passing sensitive data to functions that write to log files" id="APSC\_DV-001750-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001850" desc="Avoid passing sensitive data to functions that write to log files" id="APSC\_DV-001850-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001860" desc="Do not use weak encryption functions" id="APSC\_DV-001860-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Avoid race conditions when using fork and file descriptors" id="APSC\_DV-001995-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Avoid race conditions while checking for the existence of a symbolic link" id="APSC\_DV-001995-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Avoid race conditions while accessing files" id="APSC\_DV-001995-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Use locks to prevent race conditions when modifying bit fields" id="APSC\_DV-001995-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Do not use global variable with different locks set" id="APSC\_DV-001995-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Avoid using thread-unsafe functions" id="APSC\_DV-001995-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Usage of functions prone to race is not allowed" id="APSC\_DV-001995-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Avoid using the 'vfork()' function" id="APSC\_DV-001995-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-001995" desc="Properly define signal handlers" id="APSC\_DV-001995-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002000" desc="Ensure resources are freed" id="APSC\_DV-002000-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="APSC\_DV-002010" desc="Do not use weak encryption functions" id="APSC\_DV-002010-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002290" desc="Do not use the rand() function for generating pseudorandom numbers" id="APSC\_DV-002290-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002290" desc="Properly seed pseudorandom number generators" id="APSC\_DV-002290-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002290" desc="The 'random\_shuffle' identifier should not be used" id="APSC\_DV-002290-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002290" desc="Avoid functions which use random numbers from standard C library" id="APSC\_DV-002290-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002350" desc="Do not use weak encryption functions" id="APSC\_DV-002350-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002390" desc="Disable resolving XML external entities (XXE) in libxerces-c" id="APSC\_DV-002390-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002390" desc="Do not process structured text data natively" id="APSC\_DV-002390-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002390" desc="Do not use scanf and fscanf functions without specifying variable size in format string" id="APSC\_DV-002390-c" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="APSC\_DV-002390" desc="Do not use mbstowcs() function" id="APSC\_DV-002390-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002400" desc="Exclude unsanitized user input from format strings" id="APSC\_DV-002400-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002400" desc="The execution of a function registered with 'std::atexit()' or 'std::at\_quick\_exit()' should not exit via an exception" id="APSC\_DV-002400-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002400" desc="Avoid using the 'vfork()' function" id="APSC\_DV-002400-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002400" desc="Avoid using thread-unsafe functions" id="APSC\_DV-002400-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002440" desc="Avoid passing sensitive data to functions that write to log files" id="APSC\_DV-002440-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002460" desc="Avoid passing sensitive data to functions that write to log files" id="APSC\_DV-002460-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002470" desc="Avoid passing sensitive data to functions that write to log files" id="APSC\_DV-002470-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002480" desc="Do not print potentially sensitive information, resulting from an application error into exception messages" id="APSC\_DV-002480-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002510" desc="Protect against command injection" id="APSC\_DV-002510-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Protect against environment injection" id="APSC\_DV-002520-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Protect against file name injection" id="APSC\_DV-002520-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Protect against SQL injection" id="APSC\_DV-002520-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Never use unfiltered data from an untrusted user as the format parameter" id="APSC\_DV-002520-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Avoid tainted data in array indexes" id="APSC\_DV-002520-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Protect against integer overflow/underflow from tainted data" id="APSC\_DV-002520-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Avoid passing unvalidated binary data to log methods" id="APSC\_DV-002520-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Protect against command injection" id="APSC\_DV-002520-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Avoid printing tainted data on the output console" id="APSC\_DV-002520-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Exclude unsanitized user input from format strings" id="APSC\_DV-002520-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002520" desc="Untrusted data is used as a loop boundary" id="APSC\_DV-002520-k" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Protect against environment injection" id="APSC\_DV-002530-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Protect against file name injection" id="APSC\_DV-002530-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Protect against SQL injection" id="APSC\_DV-002530-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Never use unfiltered data from an untrusted user as the format parameter" id="APSC\_DV-002530-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Avoid tainted data in array indexes" id="APSC\_DV-002530-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Protect against integer overflow/underflow from tainted data" id="APSC\_DV-002530-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Avoid passing unvalidated binary data to log methods" id="APSC\_DV-002530-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Protect against command injection" id="APSC\_DV-002530-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Avoid printing tainted data on the output console" id="APSC\_DV-002530-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Exclude unsanitized user input from format strings" id="APSC\_DV-002530-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002530" desc="Untrusted data is used as a loop boundary" id="APSC\_DV-002530-k" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="APSC\_DV-002540" desc="Protect against SQL injection" id="APSC\_DV-002540-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Protect against environment injection" id="APSC\_DV-002550-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Protect against file name injection" id="APSC\_DV-002550-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Protect against SQL injection" id="APSC\_DV-002550-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Never use unfiltered data from an untrusted user as the format parameter" id="APSC\_DV-002550-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Avoid tainted data in array indexes" id="APSC\_DV-002550-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Protect against integer overflow/underflow from tainted data" id="APSC\_DV-002550-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Avoid passing unvalidated binary data to log methods" id="APSC\_DV-002550-g" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Protect against command injection" id="APSC\_DV-002550-h" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Avoid printing tainted data on the output console" id="APSC\_DV-002550-i" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Exclude unsanitized user input from format strings" id="APSC\_DV-002550-j" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002550" desc="Untrusted data is used as a loop boundary" id="APSC\_DV-002550-k" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Protect against environment injection" id="APSC\_DV-002560-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Protect against file name injection" id="APSC\_DV-002560-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Protect against SQL injection" id="APSC\_DV-002560-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Never use unfiltered data from an untrusted user as the format parameter" id="APSC\_DV-002560-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Avoid tainted data in array indexes" id="APSC\_DV-002560-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Protect against integer overflow/underflow from tainted data" id="APSC\_DV-002560-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Avoid passing unvalidated binary data to log methods" id="APSC\_DV-002560-g" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Protect against command injection" id="APSC\_DV-002560-h" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Avoid printing tainted data on the output console" id="APSC\_DV-002560-i" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Exclude unsanitized user input from format strings" id="APSC\_DV-002560-j" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002560" desc="Untrusted data is used as a loop boundary" id="APSC\_DV-002560-k" sev="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="APSC\_DV-002570" desc="Avoid passing sensitive data to functions that write to log files" id="APSC\_DV-002570-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002570" desc="Do not print potentially sensitive information, resulting from an application error into exception messages" id="APSC\_DV-002570-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Avoid buffer overflow due to defining incorrect format limits" id="APSC\_DV-002590-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Avoid overflow due to reading a not zero terminated string" id="APSC\_DV-002590-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Avoid overflow when reading from a buffer" id="APSC\_DV-002590-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Avoid overflow when writing to a buffer" id="APSC\_DV-002590-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Avoid integer overflows" id="APSC\_DV-002590-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Prevent buffer overflows from tainted data" id="APSC\_DV-002590-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Protect against integer overflow/underflow from tainted data" id="APSC\_DV-002590-g" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Avoid buffer overflow from tainted data due to defining incorrect format limits" id="APSC\_DV-002590-h" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Avoid buffer read overflow from tainted data" id="APSC\_DV-002590-i" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Avoid buffer write overflow from tainted data" id="APSC\_DV-002590-j" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-002590" desc="Ensure the output buffer is large enough when using path manipulation functions" id="APSC\_DV-002590-k" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-003110" desc="Do not hard code string literals" id="APSC\_DV-003110-a" sev="1">

<Stats authTot="3;" authUrg="2;" total="3"/>

</Rule>

<Rule cat="APSC\_DV-003235" desc="Avoid passing unvalidated binary data to log methods" id="APSC\_DV-003235-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="APSC\_DV-003235" desc="Avoid passing sensitive data to functions that write to log files" id="APSC\_DV-003235-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A0\_1\_1" desc="Avoid unused values" id="AUTOSAR-A0\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A0\_1\_2" desc="The value returned by a function having a non-void return type that is not an overloaded operator shall always be used" id="AUTOSAR-A0\_1\_2-a" sev="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="AUTOSAR-A0\_1\_3" desc="Every defined function with internal linkage shall be used at least once" id="AUTOSAR-A0\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A0\_1\_4" desc="Eliminate unused parameters" id="AUTOSAR-A0\_1\_4-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A0\_1\_5" desc="There shall be no unused named parameters in virtual functions" id="AUTOSAR-A0\_1\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A0\_1\_6" desc="A project shall not contain unused type declarations" id="AUTOSAR-A0\_1\_6-a" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A0\_4\_2" desc="Type long double shall not be used" id="AUTOSAR-A0\_4\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A0\_4\_4" desc="Validate values passed to library functions" id="AUTOSAR-A0\_4\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A10\_1\_1" desc="Be wary about using multiple inheritance of classes that are not abstract interfaces" id="AUTOSAR-A10\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A10\_2\_1" desc="Never redefine an inherited nonvirtual function" id="AUTOSAR-A10\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A10\_2\_1" desc="Member functions declared in derived class should not hide functions declared in base classes" id="AUTOSAR-A10\_2\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A10\_3\_1" desc="Only one of virtual, override or final should be specified in a member function declaration" id="AUTOSAR-A10\_3\_1-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A10\_3\_2" desc="Each overriding virtual function shall be declared with the override or final specifier" id="AUTOSAR-A10\_3\_2-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A10\_3\_3" desc="Do not introduce virtual functions in a final class" id="AUTOSAR-A10\_3\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A10\_3\_5" desc="A user-defined assignment operator shall not be virtual" id="AUTOSAR-A10\_3\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A10\_4\_1" desc="Hierarchies should be based on abstract classes" id="AUTOSAR-A10\_4\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A11\_0\_1" desc="A non-POD type should be defined as class" id="AUTOSAR-A11\_0\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A11\_0\_2" desc="Structs should only contain public data members and should not be a base or inherit" id="AUTOSAR-A11\_0\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A11\_3\_1" desc="Friend declarations shall not be used except declarations of comparison operators" id="AUTOSAR-A11\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_0\_1" desc="Copy and destroy consistently" id="AUTOSAR-A12\_0\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_0\_2" desc="Do not compare objects of a class that may contain padding bits with C standard library functions" id="AUTOSAR-A12\_0\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_1\_1" desc="All constructors of a class should explicitly call a constructor for all of its immediate base classes and all virtual base classes" id="AUTOSAR-A12\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_1\_1" desc="All member variables should be initialized in constructor" id="AUTOSAR-A12\_1\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_1\_2" desc="Do not specify both an NSDMI and a member initializer in a constructor for the same non-static member" id="AUTOSAR-A12\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_1\_3" desc="User-defined constructors that initialize data members with the same constant values across all constructors should initialize using NSDMI instead" id="AUTOSAR-A12\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_1\_4" desc="All constructors that are callable with a single argument of fundamental type shall be declared explicit" id="AUTOSAR-A12\_1\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_1\_5" desc="Use delegating constructors to reduce code duplication" id="AUTOSAR-A12\_1\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_1\_6" desc="Derived classes that do not need further explicit initialization and require all the constructors from the base class shall use inheriting constructors" id="AUTOSAR-A12\_1\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_4\_1" desc="Make base class destructors public and virtual, or protected and nonvirtual" id="AUTOSAR-A12\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_4\_2" desc="If a public destructor of a class is non-virtual, then the class should be declared final" id="AUTOSAR-A12\_4\_2-a" sev="4">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="AUTOSAR-A12\_6\_1" desc="Prefer initialization to assignment in constructors" id="AUTOSAR-A12\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_7\_1" desc="Define special members as =default when the behavior is equivalent to the compiler's behavior" id="AUTOSAR-A12\_7\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_8\_1" desc="A copy constructor shall only initialize its base classes and the non-static members of the class of which it is a member" id="AUTOSAR-A12\_8\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_8\_2" desc="User-defined copy and move assignment operators should use user-defined no-throw swap function" id="AUTOSAR-A12\_8\_2-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_8\_3" desc="Do not rely on the value of a moved-from object" id="AUTOSAR-A12\_8\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_8\_4" desc="Use std::move() on rvalue references and std::forward() on forwarding references" id="AUTOSAR-A12\_8\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_8\_5" desc="Check for assignment to self in operator=" id="AUTOSAR-A12\_8\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_8\_6" desc="Avoid public copy constructors and assignment operators in base classes" id="AUTOSAR-A12\_8\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A12\_8\_7" desc="Declare assignment operators with the ref-qualifier &amp;" id="AUTOSAR-A12\_8\_7-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_1\_2" desc="User defined suffixes of the user defined literal operators shall start with underscore followed by one or more letters" id="AUTOSAR-A13\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_1\_3" desc="User defined literals operators shall only perform conversion of passed parameters" id="AUTOSAR-A13\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_2\_1" desc="Have assignment operator returns a reference to \*this; make assignment operator's return type a non-const reference to it's class' type" id="AUTOSAR-A13\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_2\_2" desc="A binary arithmetic operator and a bitwise operator shall return a 'prvalue'" id="AUTOSAR-A13\_2\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_2\_3" desc="A relational operator shall return a boolean value" id="AUTOSAR-A13\_2\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_3\_1" desc="Avoid Overloading on Forwarding References" id="AUTOSAR-A13\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_5\_1" desc="When overloading the subscript operator (operator[]), implement both const and non-const versions" id="AUTOSAR-A13\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_5\_2" desc="User-conversion cast operators should be made explicit" id="AUTOSAR-A13\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_5\_3" desc="Do not use user-defined conversion functions" id="AUTOSAR-A13\_5\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_5\_4" desc="If two opposite equality operators ('==', '!=') are defined in a class, one shall be defined in terms of the other" id="AUTOSAR-A13\_5\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_5\_5" desc="Comparison operators shall be non-member functions" id="AUTOSAR-A13\_5\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_5\_5" desc="Comparison operators shall be non-member functions with identical parameter types and noexcept" id="AUTOSAR-A13\_5\_5-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A13\_6\_1" desc="Digit sequences separators ' shall only be used consistently" id="AUTOSAR-A13\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A14\_5\_1" desc="A copy constructor shall be declared when there is a template constructor with a single parameter that is a generic parameter" id="AUTOSAR-A14\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A14\_5\_3" desc="A non-member generic operator shall only be declared in a namespace that does not contain class (struct) type, enum type or union type declarations" id="AUTOSAR-A14\_5\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A14\_7\_2" desc="Template specialization shall be declared in the same file as the primary template or a user-defined type, for which the specialization is declared" id="AUTOSAR-A14\_7\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A14\_8\_2" desc="Overloaded function templates shall not be explicitly specialized" id="AUTOSAR-A14\_8\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_0\_2" desc="Ensure resources are freed" id="AUTOSAR-A15\_0\_2-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A15\_1\_1" desc="Only use instances of std::exception for exceptions" id="AUTOSAR-A15\_1\_1-a" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A15\_1\_2" desc="An exception object should not have pointer type" id="AUTOSAR-A15\_1\_2-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A15\_1\_4" desc="Ensure resources are freed" id="AUTOSAR-A15\_1\_4-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A15\_1\_5" desc="Do not throw an exception across execution boundaries" id="AUTOSAR-A15\_1\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_2\_1" desc="Function called in global or namespace scope shall not throw unhandled exceptions" id="AUTOSAR-A15\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_2\_1" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="AUTOSAR-A15\_2\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_3\_2" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="AUTOSAR-A15\_3\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_3\_3" desc="There should be at least one exception handler to catch all otherwise unhandled exceptions" id="AUTOSAR-A15\_3\_3-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A15\_3\_4" desc="Avoid using catch-all exception handlers" id="AUTOSAR-A15\_3\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_3\_5" desc="A class type exception shall always be caught by reference" id="AUTOSAR-A15\_3\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_4\_1" desc="Do not use throw exception specifications" id="AUTOSAR-A15\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_4\_2" desc="Avoid throwing exceptions from functions that are declared not to throw" id="AUTOSAR-A15\_4\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_4\_3" desc="If a function is declared with an exception-specification, then all declarations of the same function (in other translation units) shall be declared with the same set of type-ids" id="AUTOSAR-A15\_4\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_4\_4" desc="Declare functions 'noexcept' if they will not emit exceptions" id="AUTOSAR-A15\_4\_4-a" sev="2">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="AUTOSAR-A15\_4\_5" desc="Checked exceptions that could be thrown from a function shall be specified in the comment directly before the function declaration" id="AUTOSAR-A15\_4\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_1" desc="Never allow an exception to be thrown from a destructor, deallocation, and swap" id="AUTOSAR-A15\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_1" desc="All user-provided move constructors and move assignment operators shall not exit with an exception" id="AUTOSAR-A15\_5\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_2" desc="The execution of a function registered with 'std::atexit()' or 'std::at\_quick\_exit()' should not exit via an exception" id="AUTOSAR-A15\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_2" desc="The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="AUTOSAR-A15\_5\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_2" desc="The 'quick\_exit()' and '\_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used" id="AUTOSAR-A15\_5\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="The execution of a function registered with 'std::atexit()' or 'std::at\_quick\_exit()' should not exit via an exception" id="AUTOSAR-A15\_5\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="Never allow an exception to be thrown from a destructor, deallocation, and swap" id="AUTOSAR-A15\_5\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="Do not throw from within destructor" id="AUTOSAR-A15\_5\_3-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="There should be at least one exception handler to catch all otherwise unhandled exceptions" id="AUTOSAR-A15\_5\_3-d" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="An empty throw (throw;) shall only be used in the compound-statement of a catch handler" id="AUTOSAR-A15\_5\_3-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="Exceptions shall be raised only after start-up and before termination of the program" id="AUTOSAR-A15\_5\_3-f" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="AUTOSAR-A15\_5\_3-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="Where a function's declaration includes an exception-specification, the function shall only be capable of throwing exceptions of the indicated type(s)" id="AUTOSAR-A15\_5\_3-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="Function called in global or namespace scope shall not throw unhandled exceptions" id="AUTOSAR-A15\_5\_3-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="Always catch exceptions" id="AUTOSAR-A15\_5\_3-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A15\_5\_3" desc="Properly define exit handlers" id="AUTOSAR-A15\_5\_3-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A16\_0\_1" desc="The #ifndef, #ifdef, #if, #elif, #else, and #endif pre-processor directives shall only be used for conditional file inclusion and include guards" id="AUTOSAR-A16\_0\_1-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A16\_0\_1" desc="#error directive shall not be used" id="AUTOSAR-A16\_0\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A16\_0\_1" desc="The #pragma directive shall not be used" id="AUTOSAR-A16\_0\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A16\_0\_1" desc="Avoid using macro definitions" id="AUTOSAR-A16\_0\_1-d" sev="2">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="AUTOSAR-A16\_0\_1" desc="#undef shall not be used" id="AUTOSAR-A16\_0\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A16\_2\_1" desc="The following character sequences shall not appear in header file names: ', \, /\*, //, or &quot;" id="AUTOSAR-A16\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A16\_2\_2" desc="A file should directly include only the headers that contain declarations and definitions required to compile that file" id="AUTOSAR-A16\_2\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A16\_6\_1" desc="#error directive shall not be used" id="AUTOSAR-A16\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A16\_7\_1" desc="The #pragma directive shall not be used" id="AUTOSAR-A16\_7\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A17\_0\_1" desc="Do not #define nor #undef identifier 'defined'" id="AUTOSAR-A17\_0\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A17\_0\_1" desc="Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined (C99 code)" id="AUTOSAR-A17\_0\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A17\_0\_1" desc="Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined (C90 code)" id="AUTOSAR-A17\_0\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A17\_0\_1" desc="Do not redefine reserved words" id="AUTOSAR-A17\_0\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A17\_0\_1" desc="Do not #define or #undef identifiers with names which start with underscore" id="AUTOSAR-A17\_0\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A17\_1\_1" desc="The error indicator 'errno' shall not be used" id="AUTOSAR-A17\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A17\_6\_1" desc="Do not modify the standard namespaces 'std' and 'posix'" id="AUTOSAR-A17\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_0\_1" desc="The C library shall not be used" id="AUTOSAR-A18\_0\_1-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A18\_0\_2" desc="The library functions atof, atoi and atol from library stdlib.h shall not be used" id="AUTOSAR-A18\_0\_2-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A18\_0\_2" desc="The library functions atof, atoi and atol from library stdlib.h shall not be used" id="AUTOSAR-A18\_0\_2-b" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A18\_0\_3" desc="Do not include &lt;locale.h> header" id="AUTOSAR-A18\_0\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_0\_3" desc="Do not use 'setlocale' function" id="AUTOSAR-A18\_0\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_1\_1" desc="C-style arrays shall not be used" id="AUTOSAR-A18\_1\_1-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A18\_1\_2" desc="Avoid using vector&lt;bool>" id="AUTOSAR-A18\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_1\_3" desc="Prefer to use std::unique\_ptr instead of std::auto\_ptr" id="AUTOSAR-A18\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_1\_4" desc="A pointer pointing to an element of an array of objects shall not be passed to a smart pointer of single object type" id="AUTOSAR-A18\_1\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_1\_6" desc="All std::hash specializations for user-defined types shall have a noexcept function call operator" id="AUTOSAR-A18\_1\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_1" desc="Do not use calloc, malloc, realloc and free functions" id="AUTOSAR-A18\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_10" desc="Do not pass a pointer that has insufficient storage capacity or that is not suitably aligned for the object being constructed to placement 'new'" id="AUTOSAR-A18\_5\_10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_10" desc="An overhead should be used when an array of objects is passed to the placement 'new' allocation function" id="AUTOSAR-A18\_5\_10-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_11" desc="Write operator delete if you write operator new" id="AUTOSAR-A18\_5\_11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_11" desc="Write operator delete[] if you write operator new[]" id="AUTOSAR-A18\_5\_11-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_2" desc="Dynamic heap memory allocation shall not be used" id="AUTOSAR-A18\_5\_2-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_3" desc="Never provide brackets ([]) for delete when deallocating non-arrays" id="AUTOSAR-A18\_5\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_3" desc="Always provide empty brackets ([]) for delete when deallocating arrays" id="AUTOSAR-A18\_5\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_3" desc="Properly deallocate dynamically allocated resources" id="AUTOSAR-A18\_5\_3-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_4" desc="Define both sized and unsized versions of operator delete" id="AUTOSAR-A18\_5\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_5" desc="Use the same form in corresponding calls to new/malloc and delete/free" id="AUTOSAR-A18\_5\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_5" desc="Freed memory shouldn't be accessed under any circumstances" id="AUTOSAR-A18\_5\_5-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_5" desc="Properly define new handlers" id="AUTOSAR-A18\_5\_5-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_8" desc="Use allocation by declaration rather than by new or malloc" id="AUTOSAR-A18\_5\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_5\_9" desc="The user defined 'new' operator should throw the 'std::bad\_alloc' exception when the allocation fails" id="AUTOSAR-A18\_5\_9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_9\_1" desc="Prefer lambdas over std::bind, std::bind1st and std::bind2nd" id="AUTOSAR-A18\_9\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_9\_2" desc="The 'std::forward' function shall be used to forward universal references" id="AUTOSAR-A18\_9\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_9\_3" desc="Do not use std::move on objects declared with the const or const &amp; type" id="AUTOSAR-A18\_9\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A18\_9\_4" desc="Do not subsequently use the argument to std::forward" id="AUTOSAR-A18\_9\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="The 'register' storage class specifier shall not be used" id="AUTOSAR-A1\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="A copy assignment operator should be declared when a copy constructor is declared (and vice versa)" id="AUTOSAR-A1\_1\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="Both copy constructor and copy assignment operator should be declared for classes with a nontrivial destructor" id="AUTOSAR-A1\_1\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="The C library shall not be used" id="AUTOSAR-A1\_1\_1-d" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="Prefer lambdas over std::bind, std::bind1st and std::bind2nd" id="AUTOSAR-A1\_1\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="The 'binder1st' and 'binder2nd' identifiers should not be used" id="AUTOSAR-A1\_1\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="Prefer to use std::unique\_ptr instead of std::auto\_ptr" id="AUTOSAR-A1\_1\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="The 'random\_shuffle' identifier should not be used" id="AUTOSAR-A1\_1\_1-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="Do not use the increment operator (++) on an operand of type 'bool'" id="AUTOSAR-A1\_1\_1-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="The 'set\_unexpected' identifier should not be used" id="AUTOSAR-A1\_1\_1-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A1\_1\_1" desc="Do not use throw exception specifications" id="AUTOSAR-A1\_1\_1-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A20\_8\_1" desc="Do not store an already-owned pointer value in an unrelated smart pointer" id="AUTOSAR-A20\_8\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A20\_8\_2" desc="Use smart pointers when passing a pointer to an object in a thread" id="AUTOSAR-A20\_8\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A20\_8\_3" desc="Use smart pointers when passing a pointer to an object in a thread" id="AUTOSAR-A20\_8\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A20\_8\_4" desc="Consider using 'std::unique\_ptr' instead of 'std::shared\_ptr' for local objects" id="AUTOSAR-A20\_8\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A20\_8\_5" desc="'std::make\_unique' shall be used to construct objects owned by 'std::unique\_ptr'" id="AUTOSAR-A20\_8\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A20\_8\_6" desc="Prefer 'std::make\_shared' to the direct use of new" id="AUTOSAR-A20\_8\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A20\_8\_7" desc="Avoid cyclic shared\_ptr references" id="AUTOSAR-A20\_8\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A21\_8\_1" desc="Do not pass incorrect values to ctype.h library functions" id="AUTOSAR-A21\_8\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A23\_0\_1" desc="Use const container calls when the result is immediately converted to a const iterator" id="AUTOSAR-A23\_0\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A23\_0\_2" desc="Do not modify container while iterating over it" id="AUTOSAR-A23\_0\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A23\_0\_2" desc="Use valid references, pointers, and iterators to reference elements of a basic\_string" id="AUTOSAR-A23\_0\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A25\_1\_1" desc="Make predicates const pure functions" id="AUTOSAR-A25\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A25\_4\_1" desc="For associative containers never use comparison function returning true for equal values" id="AUTOSAR-A25\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A26\_5\_1" desc="Do not use the rand() function for generating pseudorandom numbers" id="AUTOSAR-A26\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A26\_5\_2" desc="Properly seed pseudorandom number generators" id="AUTOSAR-A26\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_1" desc="Avoid calling functions printf/wprintf with only one argument other than string constant" id="AUTOSAR-A27\_0\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_1" desc="Avoid tainted data in array indexes" id="AUTOSAR-A27\_0\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_1" desc="Prevent buffer overflows from tainted data" id="AUTOSAR-A27\_0\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_1" desc="Avoid buffer overflow from tainted data due to defining incorrect format limits" id="AUTOSAR-A27\_0\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_1" desc="Avoid buffer read overflow from tainted data" id="AUTOSAR-A27\_0\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_1" desc="Avoid buffer write overflow from tainted data" id="AUTOSAR-A27\_0\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_1" desc="Protect against command injection" id="AUTOSAR-A27\_0\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_1" desc="Exclude unsanitized user input from format strings" id="AUTOSAR-A27\_0\_1-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_2" desc="Avoid overflow due to reading a not zero terminated string" id="AUTOSAR-A27\_0\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_2" desc="Avoid overflow when writing to a buffer" id="AUTOSAR-A27\_0\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_2" desc="Do not use the 'char' buffer to store input from 'std::cin'" id="AUTOSAR-A27\_0\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_2" desc="Prevent buffer overflows from tainted data" id="AUTOSAR-A27\_0\_2-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_2" desc="Avoid buffer write overflow from tainted data" id="AUTOSAR-A27\_0\_2-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_3" desc="Do not alternately input and output from a stream without an intervening flush or positioning call" id="AUTOSAR-A27\_0\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_4" desc="Don't use unsafe C functions that do write to range-unchecked buffers" id="AUTOSAR-A27\_0\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_4" desc="Avoid using unsafe string functions that do not check bounds" id="AUTOSAR-A27\_0\_4-b" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_4" desc="Do not use the 'char' buffer to store input from 'std::cin'" id="AUTOSAR-A27\_0\_4-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A27\_0\_4" desc="C-style strings shall not be used" id="AUTOSAR-A27\_0\_4-d" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_1" desc="Identifier declared in a local or function prototype scope shall not hide an identifier declared in a global or namespace scope" id="AUTOSAR-A2\_10\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_1" desc="Identifiers declared in an inner local scope should not hide identifiers declared in an outer local scope" id="AUTOSAR-A2\_10\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_1" desc="Identifiers declared in a local scope should not hide identifiers declared in a class scope" id="AUTOSAR-A2\_10\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_1" desc="Identifiers declared in a class scope should not hide identifiers declared in a global or namespace scope" id="AUTOSAR-A2\_10\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_1" desc="Identifiers declared in an inner class scope should not hide identifiers declared in outer class scope" id="AUTOSAR-A2\_10\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_4" desc="The identifier name of a non-member object with static storage duration shall not be reused within a namespace" id="AUTOSAR-A2\_10\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_4" desc="The identifier name of a non-member static function shall not be reused within a namespace" id="AUTOSAR-A2\_10\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_5" desc="No object or function identifier with static storage duration should be reused" id="AUTOSAR-A2\_10\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_5" desc="No object or function identifier with static storage duration should be reused" id="AUTOSAR-A2\_10\_5-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_6" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="AUTOSAR-A2\_10\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_6" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="AUTOSAR-A2\_10\_6-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_10\_6" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="AUTOSAR-A2\_10\_6-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_11\_1" desc="Do not use the volatile keyword" id="AUTOSAR-A2\_11\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_13\_1" desc="Only those escape sequences that are defined in ISO/IEC 14882:2014 shall be used" id="AUTOSAR-A2\_13\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_13\_2" desc="String literals with different encoding prefixes shall not be concatenated" id="AUTOSAR-A2\_13\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_13\_3" desc="Type wchar\_t shall not be used" id="AUTOSAR-A2\_13\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_13\_4" desc="A string literal shall not be modified" id="AUTOSAR-A2\_13\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_13\_5" desc="Hexadecimal constants will be represented using all uppercase letters" id="AUTOSAR-A2\_13\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_13\_6" desc="Universal character names shall be used only inside character or string literals" id="AUTOSAR-A2\_13\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_3\_1" desc="Only use characters defined in ISO C standard" id="AUTOSAR-A2\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_5\_1" desc="Trigraphs shall not be used" id="AUTOSAR-A2\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_5\_1" desc="Trigraphs shall not be used" id="AUTOSAR-A2\_5\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_5\_2" desc="Do not use the following digraphs: &lt;%, %>, &lt;:, :>, %:, %:%:" id="AUTOSAR-A2\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_7\_1" desc="Line-splicing shall not be used in // comments" id="AUTOSAR-A2\_7\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_7\_2" desc="Sections of code should not be &quot;commented out&quot;" id="AUTOSAR-A2\_7\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A2\_7\_3" desc="All declarations of types, data members, and functions should be preceded by a comment annotated with the '@brief' tag" id="AUTOSAR-A2\_7\_3-a" sev="2">

<Stats authTot="33;" authUrg="0;" total="33"/>

</Rule>

<Rule cat="AUTOSAR-A2\_7\_3" desc="Function parameters and return type should be documented in a comment that precedes the function declaration" id="AUTOSAR-A2\_7\_3-b" sev="2">

<Stats authTot="32;" authUrg="0;" total="32"/>

</Rule>

<Rule cat="AUTOSAR-A2\_8\_1" desc="An include file for a class should have a file name of the form &lt;class name> + extension" id="AUTOSAR-A2\_8\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_1\_1" desc="Don't define entities with linkage in a header file" id="AUTOSAR-A3\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_1\_2" desc="Header files should have a file extension of: &quot;.h&quot;, &quot;.hpp&quot; or &quot;.hxx&quot;" id="AUTOSAR-A3\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_1\_3" desc="Implementation files in C++ will always have a file name extension of &quot;.cpp&quot;" id="AUTOSAR-A3\_1\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_1\_4" desc="When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation" id="AUTOSAR-A3\_1\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_1\_5" desc="A function definition should not be placed in a class specification unless the function is intended to be inlined" id="AUTOSAR-A3\_1\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_1\_6" desc="Trivial accessor and mutator functions should be inlined" id="AUTOSAR-A3\_1\_6-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_3\_1" desc="Objects or functions with external linkage shall be declared in a header file" id="AUTOSAR-A3\_3\_1-a" sev="2">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="AUTOSAR-A3\_3\_1" desc="Don't define entities with linkage in a header file" id="AUTOSAR-A3\_3\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_3\_2" desc="Static and thread-local objects shall be constant-initialized" id="AUTOSAR-A3\_3\_2-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A3\_8\_1" desc="Do not use resources that have been freed" id="AUTOSAR-A3\_8\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_8\_1" desc="The address of an object with automatic storage shall not be returned from a function" id="AUTOSAR-A3\_8\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_8\_1" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="AUTOSAR-A3\_8\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_8\_1" desc="Do not point to a wrapped object that has been freed" id="AUTOSAR-A3\_8\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A3\_9\_1" desc="Fixed width integer types from &lt;cstdint>, indicating the size and signedness, shall be used in place of the basic numerical types" id="AUTOSAR-A3\_9\_1-b" sev="2">

<Stats authTot="20;" authUrg="0;" total="20"/>

</Rule>

<Rule cat="AUTOSAR-A4\_10\_1" desc="NULL shall not be used as an integer value" id="AUTOSAR-A4\_10\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_10\_1" desc="Prefer 'nullptr' over 'NULL' or '0'(zero)" id="AUTOSAR-A4\_10\_1-b" sev="2">

<Stats authTot="21;" authUrg="0;" total="21"/>

</Rule>

<Rule cat="AUTOSAR-A4\_5\_1" desc="Expressions with type enum shall not be used as operands to built-in operators other than [ ], =, ==, !=, &lt;, &lt;=, >, >=, and the unary &amp; operator" id="AUTOSAR-A4\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_7\_1" desc="Avoid implicit conversions from signed to unsigned type" id="AUTOSAR-A4\_7\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_7\_1" desc="Implicit conversions from integral constant to floating type which may result in a loss of information shall not be used" id="AUTOSAR-A4\_7\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_7\_1" desc="Avoid conversions of constant values to a narrower type" id="AUTOSAR-A4\_7\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_7\_1" desc="Implicit conversions from wider to narrower integral type which may result in a loss of information shall not be used" id="AUTOSAR-A4\_7\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_7\_1" desc="Avoid implicit conversions from wider to narrower floating type" id="AUTOSAR-A4\_7\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_7\_1" desc="Avoid implicit conversions from floating to integral type" id="AUTOSAR-A4\_7\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_7\_1" desc="Implicit conversions from integral to floating type which may result in a loss of information shall not be used" id="AUTOSAR-A4\_7\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A4\_7\_1" desc="Avoid integer overflows" id="AUTOSAR-A4\_7\_1-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_1" desc="Don't write code that depends on the order of evaluation of function calls" id="AUTOSAR-A5\_0\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_1" desc="Do not use more than one volatile between two adjacent sequence points" id="AUTOSAR-A5\_0\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_1" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="AUTOSAR-A5\_0\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_1" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="AUTOSAR-A5\_0\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_1" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="AUTOSAR-A5\_0\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_1" desc="Don't write code that depends on the order of evaluation of function arguments" id="AUTOSAR-A5\_0\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_1" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="AUTOSAR-A5\_0\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_2" desc="The condition of an if-statement and the condition of an iteration-statement shall have type bool" id="AUTOSAR-A5\_0\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_3" desc="The declaration of objects should contain no more than 2 levels of pointer indirection" id="AUTOSAR-A5\_0\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_4" desc="Don't treat arrays polymorphically" id="AUTOSAR-A5\_0\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_4" desc="A pointer to an array of derived class objects should not be converted to a base class pointer" id="AUTOSAR-A5\_0\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_0\_4" desc="Do not treat arrays polymorphically" id="AUTOSAR-A5\_0\_4-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_10\_1" desc="A pointer to member virtual function shall only be tested for equality with null-pointer-constant" id="AUTOSAR-A5\_10\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_16\_1" desc="The conditional operator should not be used as a sub-expression" id="AUTOSAR-A5\_16\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_1" desc="Avoid magic numbers" id="AUTOSAR-A5\_1\_1-a" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_2" desc="Avoid default capture modes" id="AUTOSAR-A5\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_3" desc="Include a parameter list in every lambda expression" id="AUTOSAR-A5\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_4" desc="Never return lambdas that capture local objects by reference" id="AUTOSAR-A5\_1\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_4" desc="Never capture local objects from an outer lambda by reference" id="AUTOSAR-A5\_1\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_4" desc="The lambda that captures local objects by reference should not be assigned to the variable with a greater lifetime" id="AUTOSAR-A5\_1\_4-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_6" desc="Return type of a non-void return type lambda expression should be explicitly specified" id="AUTOSAR-A5\_1\_6-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_7" desc="A lambda shall not be an operand to typeid" id="AUTOSAR-A5\_1\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_1\_8" desc="Lambda expressions should not be defined inside another lambda expression" id="AUTOSAR-A5\_1\_8-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_1" desc="Avoid dynamic\_casts" id="AUTOSAR-A5\_2\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_2" desc="C-style casts (other than void casts) and functional notation casts (other than explicit constructor calls) shall not be used" id="AUTOSAR-A5\_2\_2-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_3" desc="A cast shall not remove any 'const' or 'volatile' qualification from the type of a pointer or reference" id="AUTOSAR-A5\_2\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_4" desc="Avoid using reinterpret\_cast" id="AUTOSAR-A5\_2\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_5" desc="Avoid accessing arrays out of bounds" id="AUTOSAR-A5\_2\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_5" desc="Avoid accessing arrays and pointers out of bounds" id="AUTOSAR-A5\_2\_5-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_5" desc="A pointer operand and any pointer resulting from pointer arithmetic using that operand shall both address elements of the same array" id="AUTOSAR-A5\_2\_5-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_5" desc="Avoid tainted data in array indexes" id="AUTOSAR-A5\_2\_5-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_2\_6" desc="Each operand of a logical '&amp;&amp;' or '||' shall be a postfix-expression" id="AUTOSAR-A5\_2\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_3\_1" desc="The operand of the 'typeid' operator shall not contain any expression that has side effects" id="AUTOSAR-A5\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_3\_1" desc="The operand of the 'typeid' operator shall not contain a function call that causes side effects" id="AUTOSAR-A5\_3\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_3\_2" desc="Avoid null pointer dereferencing" id="AUTOSAR-A5\_3\_2-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A5\_3\_3" desc="Do not delete objects with incomplete class at the point of deletion" id="AUTOSAR-A5\_3\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_5\_1" desc="A cast shall not convert a pointer to a function to any other pointer type, including a pointer to function type" id="AUTOSAR-A5\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A5\_6\_1" desc="Avoid division by zero" id="AUTOSAR-A5\_6\_1-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A6\_2\_1" desc="Copy assignment operators should not have side effects that could affect copying the object" id="AUTOSAR-A6\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_2\_1" desc="Move assignment operators should not have side effects that could affect moving the object" id="AUTOSAR-A6\_2\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_2\_2" desc="Expression statements shall not be explicit calls to constructors of temporary objects only" id="AUTOSAR-A6\_2\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_4\_1" desc="Every switch statement will have at least two cases and a potential default" id="AUTOSAR-A6\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_5\_1" desc="A for-loop that loops through all elements of the container and does not use its loop-counter shall not be used" id="AUTOSAR-A6\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_5\_2" desc="A for loop shall contain a single loop-counter which shall not have floating type" id="AUTOSAR-A6\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_5\_3" desc="Prefer while statements over do statements" id="AUTOSAR-A6\_5\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_5\_4" desc="The initialization expression in a for loop will perform no actions other than to initialize the value of a single for loop parameter" id="AUTOSAR-A6\_5\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_5\_4" desc="The increment expression in a for loop will perform no action other than to change a single loop parameter to the next value for the loop" id="AUTOSAR-A6\_5\_4-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A6\_6\_1" desc="The goto statement shall not be used" id="AUTOSAR-A6\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_1" desc="Declare parameters or local variable as const whenever possible" id="AUTOSAR-A7\_1\_1-a" sev="2">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_2" desc="Use constexpr to declare const variables whenever possible" id="AUTOSAR-A7\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_2" desc="Use constexpr to declare functions whenever possible" id="AUTOSAR-A7\_1\_2-b" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_3" desc="CV-qualifiers shall be placed on the right hand side of the type that is a typedef or a using name" id="AUTOSAR-A7\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_4" desc="The 'register' storage class specifier shall not be used" id="AUTOSAR-A7\_1\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_5" desc="Do not overuse 'auto' specifier" id="AUTOSAR-A7\_1\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_6" desc="Prefer alias declarations to typedefs" id="AUTOSAR-A7\_1\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_7" desc="Only one statement shall be allowed per line" id="AUTOSAR-A7\_1\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_7" desc="Multiple variable declarations shall not be allowed on the same line" id="AUTOSAR-A7\_1\_7-b" sev="2">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_7" desc="Each variable should be declared in a separate declaration statement" id="AUTOSAR-A7\_1\_7-c" sev="2">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_8" desc="Do not place type specifiers before non-type specifiers in a declaration" id="AUTOSAR-A7\_1\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_1\_9" desc="A class, structure, or enumeration will not be declared in the definition of its type" id="AUTOSAR-A7\_1\_9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_2\_1" desc="An expression with enum underlying type shall only have values corresponding to the enumerators of the enumeration" id="AUTOSAR-A7\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_2\_2" desc="Use an explicit enumeration base and ensure that it is large enough to store all enumerators" id="AUTOSAR-A7\_2\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_2\_3" desc="Prefer Scoped Enums to Unscoped Enums" id="AUTOSAR-A7\_2\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_2\_4" desc="In an enumerator list, the &quot;=&quot; construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised" id="AUTOSAR-A7\_2\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_2\_5" desc="Enumeration types shall be used instead of integer types (and constants) as case labels" id="AUTOSAR-A7\_2\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_3\_1" desc="Write a using declaration to redeclare overloaded functions" id="AUTOSAR-A7\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_4\_1" desc="Do not use the asm declaration" id="AUTOSAR-A7\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_5\_1" desc="A function shall not return a pointer or a reference to a parameter that is passed by const reference" id="AUTOSAR-A7\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_5\_2" desc="Functions shall not call themselves, either directly or indirectly" id="AUTOSAR-A7\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A7\_6\_1" desc="Never return from functions that should not return" id="AUTOSAR-A7\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_2\_1" desc="Use a trailing return type syntax if the return type is preceded by the 'typename' keyword" id="AUTOSAR-A8\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_1" desc="Do not use functions with variable numbers of arguments" id="AUTOSAR-A8\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_10" desc="A parameter shall be passed by reference if it can't be NULL" id="AUTOSAR-A8\_4\_10-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_11" desc="A smart pointer shall only be used as a parameter type if it expresses lifetime semantics" id="AUTOSAR-A8\_4\_11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_12" desc="Do not pass std::unique\_ptr by const reference" id="AUTOSAR-A8\_4\_12-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_12" desc="A smart pointer shall only be used as a parameter type if it expresses lifetime semantics" id="AUTOSAR-A8\_4\_12-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_12" desc="A parameter should only be declared as a non-const lvalue reference to 'std::shared\_ptr' or 'std::unique\_ptr' if the function replaces the managed object" id="AUTOSAR-A8\_4\_12-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_12" desc="Do not declare the type of a parameter as an rvalue reference to 'std::shared\_ptr' or 'std::unique\_ptr'" id="AUTOSAR-A8\_4\_12-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_13" desc="A smart pointer shall only be used as a parameter type if it expresses lifetime semantics" id="AUTOSAR-A8\_4\_13-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_13" desc="A parameter should only be declared as a non-const lvalue reference to 'std::shared\_ptr' or 'std::unique\_ptr' if the function replaces the managed object" id="AUTOSAR-A8\_4\_13-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_13" desc="Do not declare the type of a parameter as an rvalue reference to 'std::shared\_ptr' or 'std::unique\_ptr'" id="AUTOSAR-A8\_4\_13-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_2" desc="All exit paths from a function, except main(), with non-void return type shall have an explicit return statement with an expression" id="AUTOSAR-A8\_4\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_3" desc="Pass objects by reference instead of by value" id="AUTOSAR-A8\_4\_3-a" sev="4">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_3" desc="Declare reference parameters as const references whenever possible" id="AUTOSAR-A8\_4\_3-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_4" desc="Multiple output values from a function should be returned as a struct or tuple" id="AUTOSAR-A8\_4\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_5" desc="Use std::move() on rvalue references and std::forward() on forwarding references" id="AUTOSAR-A8\_4\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_6" desc="Use std::move() on rvalue references and std::forward() on forwarding references" id="AUTOSAR-A8\_4\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_7" desc="Pass built-in-types by value unless you are modifying them" id="AUTOSAR-A8\_4\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_7" desc="Pass small objects with a trivial copy constructor by value" id="AUTOSAR-A8\_4\_7-b" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_8" desc="Output parameters shall not be used" id="AUTOSAR-A8\_4\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_4\_9" desc="Declare reference parameters as const references whenever possible" id="AUTOSAR-A8\_4\_9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_5\_0" desc="Avoid use before initialization" id="AUTOSAR-A8\_5\_0-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_5\_1" desc="List members in an initialization list in the order in which they are declared" id="AUTOSAR-A8\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_5\_2" desc="Braced-initialization {}, without equals sign, shall be used for variable initialization" id="AUTOSAR-A8\_5\_2-a" sev="2">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="AUTOSAR-A8\_5\_3" desc="A variable of type auto shall not be initialized using '{}' or '={}' braced-initialization" id="AUTOSAR-A8\_5\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A8\_5\_4" desc="Avoid overloading constructors with std::initializer\_list" id="AUTOSAR-A8\_5\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A9\_3\_1" desc="Public member functions shall not return non-const handles to private/protected class-data" id="AUTOSAR-A9\_3\_1-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-A9\_3\_1" desc="Protected member function shall not return non-const handles to private class-data" id="AUTOSAR-A9\_3\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A9\_5\_1" desc="Unions shall not be used" id="AUTOSAR-A9\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A9\_5\_1" desc="Unions shall not be used" id="AUTOSAR-A9\_5\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A9\_6\_1" desc="Bit fields shall only be defined to be of type unsigned int or signed int" id="AUTOSAR-A9\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-A9\_6\_2" desc="Do not declare member variables as bit-fields" id="AUTOSAR-A9\_6\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_1" desc="There shall be no unreachable code in &quot;if/else/while/for&quot; block" id="AUTOSAR-M0\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_1" desc="There shall be no unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="AUTOSAR-M0\_1\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_1" desc="There shall be no unreachable code in &quot;else&quot; block" id="AUTOSAR-M0\_1\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_1" desc="There shall be no unreachable code after &quot;if&quot; or &quot;switch&quot; statement inside while/for/do...while loop" id="AUTOSAR-M0\_1\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_1" desc="There shall be no unreachable code after 'if' or 'switch' statement" id="AUTOSAR-M0\_1\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_1" desc="There shall be no unreachable code in 'for' loop" id="AUTOSAR-M0\_1\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_1" desc="There shall be no unreachable code in switch statement" id="AUTOSAR-M0\_1\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_10" desc="Every defined function with external linkage shall be used at least once" id="AUTOSAR-M0\_1\_10-a" sev="4">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_10" desc="Every defined function with internal linkage shall be used at least once" id="AUTOSAR-M0\_1\_10-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-aa" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-ab" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Avoid conditions that always evaluate to the same value" id="AUTOSAR-M0\_1\_2-ac" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-m" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-n" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-o" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-p" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-q" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-r" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-s" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-t" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-u" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-v" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-w" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-x" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-y" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_2" desc="Boolean operations whose results are invariant shall not be permitted" id="AUTOSAR-M0\_1\_2-z" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_3" desc="Avoid unused local variables" id="AUTOSAR-M0\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_3" desc="Avoid unnecessary local variables" id="AUTOSAR-M0\_1\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_3" desc="Avoid unused private member variables" id="AUTOSAR-M0\_1\_3-c" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_4" desc="A project shall not contain non-volatile POD variables having only one use" id="AUTOSAR-M0\_1\_4-a" sev="2">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_8" desc="All non-empty functions with void return type shall have external side effect(s)" id="AUTOSAR-M0\_1\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_8" desc="Functions with void return type shall not be empty" id="AUTOSAR-M0\_1\_8-b" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-M0\_1\_9" desc="All non-null statements shall either have at least one side-effect however executed or cause control flow to change" id="AUTOSAR-M0\_1\_9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_2\_1" desc="An object shall not be assigned to an overlapping object" id="AUTOSAR-M0\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_2\_1" desc="An object shall not be assigned to an overlapping object" id="AUTOSAR-M0\_2\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Do not subtract two pointers that do not address elements of the same array" id="AUTOSAR-M0\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Avoid overflow when writing to a buffer" id="AUTOSAR-M0\_3\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Do not compare two unrelated pointers" id="AUTOSAR-M0\_3\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Avoid accessing arrays out of bounds" id="AUTOSAR-M0\_3\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Avoid division by zero" id="AUTOSAR-M0\_3\_1-e" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Avoid null pointer dereferencing" id="AUTOSAR-M0\_3\_1-f" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Avoid overflow due to reading a not zero terminated string" id="AUTOSAR-M0\_3\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Avoid buffer overflow due to defining incorrect format limits" id="AUTOSAR-M0\_3\_1-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Avoid overflow when reading from a buffer" id="AUTOSAR-M0\_3\_1-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_1" desc="Do not check for null after dereferencing" id="AUTOSAR-M0\_3\_1-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M0\_3\_2" desc="If a function returns error information, then that error information shall be tested" id="AUTOSAR-M0\_3\_2-a" sev="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="AUTOSAR-M0\_4\_2" desc="Use of floating-point arithmetic shall be documented" id="AUTOSAR-M0\_4\_2-a" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="AUTOSAR-M10\_1\_1" desc="Classes should not be derived from virtual bases" id="AUTOSAR-M10\_1\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M10\_1\_2" desc="A base class shall only be declared virtual if it is used in a diamond hierarchy" id="AUTOSAR-M10\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M10\_1\_3" desc="A base class shall not be both virtual and non-virtual in the same hierarchy" id="AUTOSAR-M10\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M10\_2\_1" desc="All accessible entity names within a multiple inheritance hierarchy should be unique" id="AUTOSAR-M10\_2\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M10\_3\_3" desc="A virtual function shall only be overridden by a pure virtual function if it is itself declared as pure virtual" id="AUTOSAR-M10\_3\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M11\_0\_1" desc="Member data in non-POD types shall be private" id="AUTOSAR-M11\_0\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M12\_1\_1" desc="Do not use dynamic type of an object under construction" id="AUTOSAR-M12\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M14\_5\_3" desc="A copy assignment operator shall be declared when there is a template assignment operator with a parameter that is a generic parameter" id="AUTOSAR-M14\_5\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M14\_6\_1" desc="In a class template with a dependent base, any name that may be found in that dependent base shall be referred to using a qualified-id or this->" id="AUTOSAR-M14\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_0\_3" desc="Control shall not be transferred into a try or catch block using a goto or a switch statement" id="AUTOSAR-M15\_0\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_1\_1" desc="The assignment-expression of a throw statement shall not itself cause an exception to be thrown" id="AUTOSAR-M15\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_1\_2" desc="NULL shall not be thrown explicitly" id="AUTOSAR-M15\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_1\_3" desc="An empty throw (throw;) shall only be used in the compound-statement of a catch handler" id="AUTOSAR-M15\_1\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_3\_1" desc="Exceptions shall be raised only after start-up and before termination of the program" id="AUTOSAR-M15\_3\_1-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-M15\_3\_3" desc="Handlers of a function-try-block implementation of a class constructor or destructor shall not reference nonstatic members from this class or its bases" id="AUTOSAR-M15\_3\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_3\_4" desc="Function called in global or namespace scope shall not throw unhandled exceptions" id="AUTOSAR-M15\_3\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_3\_4" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="AUTOSAR-M15\_3\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_3\_6" desc="Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class" id="AUTOSAR-M15\_3\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M15\_3\_7" desc="Where multiple handlers are provided in a single 'try-catch' statement or 'function-try-block', any ellipsis (catch-all) handler shall occur last" id="AUTOSAR-M15\_3\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_0\_1" desc="#include statements in a file should only be preceded by other preprocessor directives or comments" id="AUTOSAR-M16\_0\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_0\_2" desc="Macros shall not be #define'd or #undef'd within a block" id="AUTOSAR-M16\_0\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_0\_5" desc="Arguments to a function-like macro shall not contain tokens that look like preprocessing directives" id="AUTOSAR-M16\_0\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_0\_6" desc="In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##" id="AUTOSAR-M16\_0\_6-a" sev="2">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="AUTOSAR-M16\_0\_7" desc="Do not use in preprocessor directives #if and #elif macros not defined in translation unit" id="AUTOSAR-M16\_0\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_0\_8" desc="Preprocessing directives shall be syntactically meaningful even when excluded by the preprocessor" id="AUTOSAR-M16\_0\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_1\_1" desc="The defined preprocessor operator shall only be used in one of the two standard forms" id="AUTOSAR-M16\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_1\_2" desc="All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if or #ifdef directive to which they are related" id="AUTOSAR-M16\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_2\_3" desc="Use multiple include guards" id="AUTOSAR-M16\_2\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_3\_1" desc="There shall be at most one occurrence of the # or ## preprocessor operators in a single macro definition" id="AUTOSAR-M16\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M16\_3\_2" desc="The # and ## preprocessor operators should not be used" id="AUTOSAR-M16\_3\_2-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M17\_0\_2" desc="The names of standard library macros and objects shall not be reused" id="AUTOSAR-M17\_0\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M17\_0\_3" desc="The names of standard library functions shall not be overridden" id="AUTOSAR-M17\_0\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M17\_0\_5" desc="The setjmp macro and the longjmp function shall not be used" id="AUTOSAR-M17\_0\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_0\_3" desc="The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="AUTOSAR-M18\_0\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_0\_3" desc="The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="AUTOSAR-M18\_0\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_0\_3" desc="The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="AUTOSAR-M18\_0\_3-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_0\_3" desc="The 'getenv()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="AUTOSAR-M18\_0\_3-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_0\_4" desc="The time handling functions of library time.h shall not be used" id="AUTOSAR-M18\_0\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_0\_5" desc="The unbounded functions of library &lt;cstring> shall not be used" id="AUTOSAR-M18\_0\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_2\_1" desc="The macro offsetof, in library stddef.h, shall not be used" id="AUTOSAR-M18\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_7\_1" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="AUTOSAR-M18\_7\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M18\_7\_1" desc="The standard header file &lt;signal.h> shall not be used" id="AUTOSAR-M18\_7\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M19\_3\_1" desc="The error indicator 'errno' shall not be used" id="AUTOSAR-M19\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M27\_0\_1" desc="The input/output library stdio.h shall not be used" id="AUTOSAR-M27\_0\_1-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-M2\_10\_1" desc="Different identifiers shall be typographically unambiguous" id="AUTOSAR-M2\_10\_1-a" sev="2">

<Stats authTot="13;" authUrg="0;" total="13"/>

</Rule>

<Rule cat="AUTOSAR-M2\_13\_2" desc="Octal constants (other than zero) shall not be used" id="AUTOSAR-M2\_13\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M2\_13\_2" desc="Octal escape sequences shall not be used" id="AUTOSAR-M2\_13\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M2\_13\_3" desc="A &quot;U&quot; suffix shall be applied to all octal or hexadecimal integer literals of unsigned type" id="AUTOSAR-M2\_13\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M2\_13\_4" desc="Literal suffixes shall use uppercase rather than lowercase letters" id="AUTOSAR-M2\_13\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M2\_7\_1" desc="The character sequence /\* shall not be used within a C-style comment" id="AUTOSAR-M2\_7\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_1\_2" desc="Always declare functions at file scope" id="AUTOSAR-M3\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_2\_1" desc="All declarations of an object or function shall have compatible types" id="AUTOSAR-M3\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_2\_2" desc="The One Definition Rule shall not be violated" id="AUTOSAR-M3\_2\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_2\_3" desc="A type, object or function that is used in multiple translation units shall be declared in one and only one file" id="AUTOSAR-M3\_2\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_2\_4" desc="An identifier with external linkage shall have exactly one external definition" id="AUTOSAR-M3\_2\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_3\_2" desc="If a function has internal linkage then all re-declarations shall include the 'static' storage class specifier" id="AUTOSAR-M3\_3\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_4\_1" desc="Declare variables as locally as possible" id="AUTOSAR-M3\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_4\_1" desc="Objects shall be defined at block scope if they are only accessed from within a single function" id="AUTOSAR-M3\_4\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_9\_1" desc="The types used for an object, a function return type, or a function parameter shall be token-for-token identical in all declarations and re-declarations" id="AUTOSAR-M3\_9\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M3\_9\_3" desc="The underlying bit representations of floating-point values shall not be used" id="AUTOSAR-M3\_9\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M4\_10\_1" desc="NULL shall not be used as an integer value" id="AUTOSAR-M4\_10\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M4\_10\_2" desc="Literal zero (0) shall not be used as the null-pointer-constant" id="AUTOSAR-M4\_10\_2-a" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="AUTOSAR-M4\_5\_1" desc="Expressions that are effectively Boolean should not be used as operands to operators other than (&amp;&amp;, ||, !, =, ==, !=, ?:)" id="AUTOSAR-M4\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M4\_5\_3" desc="Expressions with type (plain) char and wchar\_t shall not be used as operands to built-in operators other than =, ==, != and the unary &amp; operator" id="AUTOSAR-M4\_5\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_10" desc="If the bitwise operators ~ and &lt;&lt; are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand" id="AUTOSAR-M5\_0\_10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_11" desc="The plain char type shall be used only for the storage and use of character values" id="AUTOSAR-M5\_0\_11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_12" desc="signed and unsigned char type shall be used only for the storage and use of numeric values" id="AUTOSAR-M5\_0\_12-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_14" desc="The first operand of a conditional-operator shall have type bool" id="AUTOSAR-M5\_0\_14-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_15" desc="Array indexing shall be the only allowed form of pointer arithmetic" id="AUTOSAR-M5\_0\_15-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_16" desc="Avoid accessing arrays out of bounds" id="AUTOSAR-M5\_0\_16-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_16" desc="A pointer operand and any pointer resulting from pointer arithmetic using that operand shall both address elements of the same array" id="AUTOSAR-M5\_0\_16-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_17" desc="Do not subtract two pointers that do not address elements of the same array" id="AUTOSAR-M5\_0\_17-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_18" desc="Do not compare two unrelated pointers" id="AUTOSAR-M5\_0\_18-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_2" desc="Use parenthesis for the right-hand operand of an assignment operator when it contains an assignment expression" id="AUTOSAR-M5\_0\_2-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_2" desc="Limited dependence should be placed on C's operator precedence rules in expressions" id="AUTOSAR-M5\_0\_2-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_2" desc="No parentheses are required for the operand of a unary operator" id="AUTOSAR-M5\_0\_2-c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_2" desc="Limited dependence should be placed on C's operator precedence rules in expressions" id="AUTOSAR-M5\_0\_2-d" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_2" desc="Use parentheses unless all operators in the expression are the same" id="AUTOSAR-M5\_0\_2-e" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_2" desc="Limited dependence should be placed on C's operator precedence rules in expressions" id="AUTOSAR-M5\_0\_2-f" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_20" desc="Non-constant operands to a binary bitwise operator shall have the same underlying type" id="AUTOSAR-M5\_0\_20-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_21" desc="Bitwise operators shall only be applied to operands of unsigned underlying type" id="AUTOSAR-M5\_0\_21-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_3" desc="Avoid implicit conversions of complex expressions" id="AUTOSAR-M5\_0\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_3" desc="Avoid implicit conversions of complex expressions" id="AUTOSAR-M5\_0\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_3" desc="Avoid implicit conversions of complex expressions" id="AUTOSAR-M5\_0\_3-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_4" desc="Avoid implicit conversions between signed and unsigned integer types" id="AUTOSAR-M5\_0\_4-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_5" desc="There shall be no implicit conversions from integral to floating type" id="AUTOSAR-M5\_0\_5-a" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_5" desc="Avoid implicit conversions from floating to integral type" id="AUTOSAR-M5\_0\_5-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_6" desc="Avoid implicit integral conversions from a wider to a narrower type" id="AUTOSAR-M5\_0\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_6" desc="Avoid implicit conversions from wider to narrower floating type" id="AUTOSAR-M5\_0\_6-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_6" desc="Avoid implicit conversions of floating point numbers from wider to narrower floating type" id="AUTOSAR-M5\_0\_6-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_7" desc="The value of a complex expression of floating type should not be cast to an integer type" id="AUTOSAR-M5\_0\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_7" desc="The value of a complex expression of integer type shall only be cast to a type of the same signedness that is no wider than the underlying type of the expression" id="AUTOSAR-M5\_0\_7-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_8" desc="The value of a complex expression of integer type shall only be cast to a type of the same signedness that is no wider than the underlying type of the expression" id="AUTOSAR-M5\_0\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_8" desc="The value of a complex expression of floating type should not be cast to a wider floating type" id="AUTOSAR-M5\_0\_8-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_0\_9" desc="The value of a complex expression of integer type shall only be cast to a type of the same signedness that is no wider than the underlying type of the expression" id="AUTOSAR-M5\_0\_9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_14\_1" desc="The right-hand operand of a logical &amp;&amp; or || operator shall not contain side effects" id="AUTOSAR-M5\_14\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_17\_1" desc="The overloaded binary operator should be implemented in terms of its corresponding compound assignment operator" id="AUTOSAR-M5\_17\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_18\_1" desc="The comma operator shall not be used" id="AUTOSAR-M5\_18\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_19\_1" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="AUTOSAR-M5\_19\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_19\_1" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="AUTOSAR-M5\_19\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_10" desc="The increment (++) and decrement (--) operators should not be mixed with other operators in an expression" id="AUTOSAR-M5\_2\_10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_11" desc="Avoid overloading comma operator &quot;,&quot;" id="AUTOSAR-M5\_2\_11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_11" desc="Avoid overloading logical operators AND, OR (&amp;&amp;, ||)" id="AUTOSAR-M5\_2\_11-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_12" desc="An identifier with array type passed as a function argument shall not decay to a pointer" id="AUTOSAR-M5\_2\_12-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_2" desc="A pointer to a virtual base class shall only be cast to a pointer to a derived class by means of dynamic\_cast" id="AUTOSAR-M5\_2\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_3" desc="Casts from a base class to a derived class should not be performed on polymorphic types" id="AUTOSAR-M5\_2\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_6" desc="A cast shall not convert a pointer to a function to any other pointer type, including a pointer to function type" id="AUTOSAR-M5\_2\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_8" desc="An object with integer type or pointer to void type shall not be converted to an object with pointer type" id="AUTOSAR-M5\_2\_8-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="AUTOSAR-M5\_2\_9" desc="A cast should not convert a pointer type to an integral type" id="AUTOSAR-M5\_2\_9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_3\_1" desc="Each operand of the ! operator, the logical &amp;&amp; or the logical || operators shall have type bool" id="AUTOSAR-M5\_3\_1-a" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="AUTOSAR-M5\_3\_2" desc="The unary minus operator shall not be applied to an expression whose underlying type is unsigned" id="AUTOSAR-M5\_3\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_3\_3" desc="The unary &amp; operator shall not be overloaded" id="AUTOSAR-M5\_3\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_3\_4" desc="The operand of the sizeof operator shall not contain any expression which has side effects" id="AUTOSAR-M5\_3\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_3\_4" desc="Object designated by a volatile lvalue should not be accessed in the operand of the sizeof operator" id="AUTOSAR-M5\_3\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_3\_4" desc="The function call that causes the side effect shall not be the operand of the sizeof operator" id="AUTOSAR-M5\_3\_4-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M5\_8\_1" desc="The right-hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left-hand operand" id="AUTOSAR-M5\_8\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_2\_1" desc="Assignment operators shall not be used in expressions that yield a Boolean value" id="AUTOSAR-M6\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_2\_2" desc="Floating-point expressions shall not be tested for equality or inequality" id="AUTOSAR-M6\_2\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_2\_3" desc="A null statement shall occur on a line by itself or be followed by a comment" id="AUTOSAR-M6\_2\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_3\_1" desc="The statement forming the body of a 'switch', 'while', 'do...while' or 'for' statement shall be a compound statement" id="AUTOSAR-M6\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_1" desc="'if' and 'else' should be followed by a compound statement" id="AUTOSAR-M6\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_2" desc="All 'if...else-if' constructs shall be terminated with an 'else' clause" id="AUTOSAR-M6\_4\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_3" desc="An unconditional throw or break statement shall terminate every non-empty switch-clause" id="AUTOSAR-M6\_4\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_3" desc="The final clause of a switch statement shall be the default-clause unless all enumeration values are tested" id="AUTOSAR-M6\_4\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_3" desc="A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="AUTOSAR-M6\_4\_3-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_3" desc="Every switch statement shall have at least one case clause" id="AUTOSAR-M6\_4\_3-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_3" desc="A switch expression shall not represent a value that is effectively Boolean" id="AUTOSAR-M6\_4\_3-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_4" desc="A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="AUTOSAR-M6\_4\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_5" desc="An unconditional throw or break statement shall terminate every non-empty switch-clause" id="AUTOSAR-M6\_4\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_6" desc="The final clause of a switch statement shall be the default-clause unless all enumeration values are tested" id="AUTOSAR-M6\_4\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_4\_7" desc="A switch expression shall not represent a value that is effectively Boolean" id="AUTOSAR-M6\_4\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_5\_2" desc="If loop-counter is not modified by -- or ++, then, within condition, the loop-counter shall only be used as an operand to &lt;=, &lt;, > or >=" id="AUTOSAR-M6\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_5\_3" desc="Do not modify for loop counter within a body of the loop" id="AUTOSAR-M6\_5\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_5\_4" desc="The loop-counter shall be modified by one of: --, ++, -=n, or +=n; where n remains constant for the duration of the loop" id="AUTOSAR-M6\_5\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_5\_5" desc="A loop-control-variable other than the loop-counter shall not be modified within condition or expression" id="AUTOSAR-M6\_5\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_5\_6" desc="A loop-control-variable other than the loop-counter which is modified in statement shall have type bool" id="AUTOSAR-M6\_5\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_6\_1" desc="Any label referenced by a goto statement shall be declared in the same block, or in a block enclosing the goto statement" id="AUTOSAR-M6\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_6\_2" desc="The goto statement shall jump to a label declared later in the same function body" id="AUTOSAR-M6\_6\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M6\_6\_3" desc="The continue statement shall only be used within a well formed for loop" id="AUTOSAR-M6\_6\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_1\_2" desc="Declare a type of parameter as typedef to pointer to const if the pointer is not used to modify the addressed object" id="AUTOSAR-M7\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_1\_2" desc="A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object" id="AUTOSAR-M7\_1\_2-b" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="AUTOSAR-M7\_1\_2" desc="Declare reference parameters as const references whenever possible" id="AUTOSAR-M7\_1\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_3\_1" desc="The global namespace shall only contain main() and namespace declarations" id="AUTOSAR-M7\_3\_1-a" sev="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="AUTOSAR-M7\_3\_2" desc="The identifier main shall not be used for a function other than the global function main" id="AUTOSAR-M7\_3\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_3\_3" desc="There shall be no unnamed namespaces in header files" id="AUTOSAR-M7\_3\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_3\_4" desc="using-directives shall not be used" id="AUTOSAR-M7\_3\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_3\_6" desc="using-directives and using-declarations (excluding class scope or function scope using-declarations) shall not be used in header files" id="AUTOSAR-M7\_3\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_4\_1" desc="All usage of assembler shall be documented" id="AUTOSAR-M7\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_4\_2" desc="Assembler instructions shall only be introduced using the asm declaration" id="AUTOSAR-M7\_4\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_4\_3" desc="Assembly language shall be encapsulated and isolated in C/C++ functions" id="AUTOSAR-M7\_4\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_5\_1" desc="The address of an object with automatic storage shall not be returned from a function" id="AUTOSAR-M7\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M7\_5\_2" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="AUTOSAR-M7\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M8\_0\_1" desc="Each variable should be declared in a separate declaration statement" id="AUTOSAR-M8\_0\_1-a" sev="2">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="AUTOSAR-M8\_3\_1" desc="Do not redefine an inherited virtual function with a different default parameter value" id="AUTOSAR-M8\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M8\_4\_2" desc="The identifiers used in the declaration and definition of a function shall be identical" id="AUTOSAR-M8\_4\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M8\_4\_4" desc="A function identifier shall only be used with either a preceding '&amp;', or with a parenthesised parameter list, which may be empty" id="AUTOSAR-M8\_4\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M8\_5\_2" desc="The initializer for an aggregate or union shall be enclosed in braces" id="AUTOSAR-M8\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M8\_5\_2" desc="Arrays shall not be partially initialized" id="AUTOSAR-M8\_5\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M8\_5\_2" desc="The non-zero initialization of structures requires an explicit initializer for each element" id="AUTOSAR-M8\_5\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M9\_3\_1" desc="Const member functions shall not return non-const pointers or references to class-data" id="AUTOSAR-M9\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="AUTOSAR-M9\_3\_3" desc="If a member function can be made static then it shall be made static, otherwise if it can be made const then it shall be made const" id="AUTOSAR-M9\_3\_3-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="AUTOSAR-M9\_6\_4" desc="Named bit-fields with signed integer type shall have a length of more than one bit" id="AUTOSAR-M9\_6\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-API" desc="Do not pass incorrect values to library functions" id="BD-API-BADPARAM" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-API" desc="Do not pass negative values to functions expecting non-negative arguments" id="BD-API-NEGPARAM" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-API" desc="Do not pass incorrect values to ctype.h library functions" id="BD-API-CTYPE" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-API" desc="Do not check for a non-negative value after passing as a non-negative argument" id="BD-API-REVNEGPARAM" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-API" desc="The size\_t argument passed to any function in string.h shall have an appropriate value" id="BD-API-STRSIZE" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-API" desc="Validate values passed to library functions" id="BD-API-VALPARAM" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-CO" desc="Do not access iterator out of range" id="BD-CO-ITOUT" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-CO" desc="Use valid references, pointers, and iterators to reference elements of a basic\_string" id="BD-CO-STRMOD" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-CO" desc="Do not pass empty container iterators to std algorithms as destinations" id="BD-CO-EMPCON" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-CO" desc="Do not compare iterators from different containers" id="BD-CO-ITINVCOMP" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-CO" desc="Do not modify container while iterating over it" id="BD-CO-ITMOD" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-MISC" desc="Always close transactions" id="BD-MISC-TRANS" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-MISC" desc="Prevent calling unsafe functions (custom rule)" id="BD-MISC-DC" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Always catch exceptions" id="BD-PB-EXCEPT" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not modify string literals" id="BD-PB-MODSTR" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid use before initialization" id="BD-PB-NOTINIT" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid null pointer dereferencing" id="BD-PB-NP" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid buffer overflow due to defining incorrect format limits" id="BD-PB-OVERFFMT" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid overflow due to reading a not zero terminated string" id="BD-PB-OVERFNZT" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid overflow when reading from a buffer" id="BD-PB-OVERFRD" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid overflow when writing to a buffer" id="BD-PB-OVERFWR" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid overwriting method parameters" id="BD-PB-POVR" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not point to a wrapped object that has been freed" id="BD-PB-WRAPESC" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid division by zero" id="BD-PB-ZERO" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid accessing arrays out of bounds" id="BD-PB-ARRAY" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="A pointer pointing to an element of an array of objects shall not be passed to a smart pointer of single object type" id="BD-PB-ARRPTR" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid incorrect shift operations" id="BD-PB-BADSHIFT" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Use the correct byte ordering when transferring data between systems" id="BD-PB-BYTEORD" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid conditions that always evaluate to the same value" id="BD-PB-CC" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Consistently check the returned value of non-void functions" id="BD-PB-CHECKRET" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Always check the returned value of non-void function" id="BD-PB-CHECKRETGEN" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not check for null after dereferencing" id="BD-PB-DEREF" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not modify the alignment of objects by calling realloc()" id="BD-PB-DNMPTR" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="The macro EOF should be compared with the unmodified return value from the Standard Library function" id="BD-PB-EOFCOMP" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Properly use errno value" id="BD-PB-ERRNO" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Properly define exit handlers" id="BD-PB-EXITHAN" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Reset strings on fgets() or fgetws() failure" id="BD-PB-FGETS" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Only use values for fsetpos() that are returned from fgetpos()" id="BD-PB-FSETPOS" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid infinite recursion" id="BD-PB-INFREC" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not rely on an environment pointer following an operation that may invalidate it" id="BD-PB-INVENV" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Pointers returned by certain Standard Library functions should not be used following a subsequent call to the same or related function" id="BD-PB-INVRET" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="The Standard Library function memcmp shall not be used to compare null terminated strings" id="BD-PB-MCCSTR" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid calls to memory-setting functions that can be optimized out by the compiler" id="BD-PB-MEMOPT" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Properly define new handlers" id="BD-PB-NEWHAN" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid throwing exceptions from functions that are declared not to throw" id="BD-PB-NOEXCEPT" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Declare non-returning functions with the attribute that specifies that they do not return" id="BD-PB-NORETDECL" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Never return from functions that should not return" id="BD-PB-NORETURN" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid accessing arrays and pointers out of bounds" id="BD-PB-OVERFARRAY" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="An object shall not be assigned or copied to an overlapping object" id="BD-PB-OVERLAP" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not treat arrays polymorphically" id="BD-PB-POLARR" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="A pointer operand and any pointer resulting from pointer arithmetic using that operand shall both address elements of the same array" id="BD-PB-PTRARR" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not compare two unrelated pointers" id="BD-PB-PTRCMP" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not subtract two pointers that do not address elements of the same array" id="BD-PB-PTRSUB" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not store an already-owned pointer value in an unrelated smart pointer" id="BD-PB-PTRVALUE" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not call putenv() with a pointer to an automatic variable as the argument" id="BD-PB-PUTENV" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Properly define signal handlers" id="BD-PB-SIGHAN" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not return from a computational exception signal handler" id="BD-PB-SIGRETURN" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not alternately input and output from a stream without an intervening flush or positioning call" id="BD-PB-STREAMINOUT" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not subsequently use the argument to std::forward" id="BD-PB-SUBSEQFRWD" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not rely on the value of a moved-from object" id="BD-PB-SUBSEQMOVE" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid switch with unreachable branches" id="BD-PB-SWITCH" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Properly define terminate handlers" id="BD-PB-TERMHAN" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Properly define unexpected handlers" id="BD-PB-UNEXPHAN" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not use dynamic type of an object under construction" id="BD-PB-VCTOR" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not use dynamic type of an object under destruction" id="BD-PB-VDTOR" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Ensure the size of the variable length array is in valid range" id="BD-PB-VLASIZE" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="The same file shall not be opened for read and write access at the same time on different streams" id="BD-PB-WRRDSTR" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid writing to a stream which has been opened as read only" id="BD-PB-WRROS" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid integer overflows" id="BD-PB-INTOVERF" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Ensure strings are zero terminated" id="BD-PB-NZTS" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Ensure the output buffer is large enough when using path manipulation functions" id="BD-PB-PATHBUF" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="A parameter shall be passed by reference if it can't be NULL" id="BD-PB-REFPARAM" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="BD-PB" desc="Always throw the created std::exception object" id="BD-PB-STDEXC" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Suspicious setting of stream flags" id="BD-PB-STREAMFLAGS" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Restore stream format" id="BD-PB-STREAMFMT" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Use macros for variable arguments correctly" id="BD-PB-VARARGS" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid unused values" id="BD-PB-VOVR" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Avoid unreachable methods" id="BD-PB-UCMETH" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Guarantee that container indices are within the valid range" id="BD-PB-VALRANGE" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-PB" desc="Do not use recursion" id="BD-PB-RECFUN" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-RES" desc="Properly deallocate dynamically allocated resources" id="BD-RES-BADDEALLOC" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-RES" desc="Ensure deallocation functions guarantee resource freeing" id="BD-RES-BADFREEF" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-RES" desc="Do not use resources that have been freed" id="BD-RES-FREE" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-RES" desc="Allocate sufficient memory to hold an object of a given type" id="BD-RES-INSUFMEM" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-RES" desc="Do not free resources using invalid pointers" id="BD-RES-INVFREE" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-RES" desc="Ensure resources are freed" id="BD-RES-LEAKS" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="BD-RES" desc="Avoid cyclic shared\_ptr references" id="BD-RES-CSP" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-RES" desc="Do not create variables on the stack above the defined limits" id="BD-RES-STACKLIM" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Avoid tainted data in array indexes" id="BD-SECURITY-ARRAY" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Prevent buffer overflows from tainted data" id="BD-SECURITY-BUFWRITE" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Protect against integer overflow/underflow from tainted data" id="BD-SECURITY-INTOVERF" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Avoid buffer overflow from tainted data due to defining incorrect format limits" id="BD-SECURITY-OVERFFMT" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Avoid buffer read overflow from tainted data" id="BD-SECURITY-OVERFRD" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Avoid buffer write overflow from tainted data" id="BD-SECURITY-OVERFWR" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Protect against command injection" id="BD-SECURITY-TDCMD" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Avoid printing tainted data on the output console" id="BD-SECURITY-TDCONSOLE" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Protect against environment injection" id="BD-SECURITY-TDENV" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Protect against file name injection" id="BD-SECURITY-TDFNAMES" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Exclude unsanitized user input from format strings" id="BD-SECURITY-TDINPUT" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Validate potentially tainted data before it is used in the controlling expression of a loop" id="BD-SECURITY-TDLOOP" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Protect against SQL injection" id="BD-SECURITY-TDSQL" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Avoid passing unvalidated binary data to log methods" id="BD-SECURITY-LOG" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Properly seed pseudorandom number generators" id="BD-SECURITY-RAND" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Sensitive data should be cleared before being deallocated" id="BD-SECURITY-SENSFREE" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Avoid passing sensitive data to functions that write to log files" id="BD-SECURITY-SENSLOG" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Validate potentially tainted data before it is used to determine the size of memory allocation" id="BD-SECURITY-TDALLOC" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="BD-SECURITY" desc="Disable resolving XML external entities (XXE) in libxerces-c" id="BD-SECURITY-XXEXRC" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Avoid double locking" id="BD-TRS-DLOCK" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Avoid race conditions when using fork and file descriptors" id="BD-TRS-FORKFILE" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Do not abandon unreleased locks" id="BD-TRS-LOCK" sev="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="BD-TRS" desc="Do not use global variable with different locks set" id="BD-TRS-MLOCK" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Do not acquire locks in different order" id="BD-TRS-ORDER" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Avoid race conditions while checking for the existence of a symbolic link" id="BD-TRS-SYMLINK" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Do not use blocking functions while holding a lock" id="BD-TRS-TSHL" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Declare objects shared between POSIX threads with appropriate storage durations" id="BD-TRS-ARG" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Make const member functions thread-safe" id="BD-TRS-CMF" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Variable should be used in context of single critical section" id="BD-TRS-DIFCS" sev="2">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="BD-TRS" desc="Avoid race conditions while accessing files" id="BD-TRS-FRC" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Do not join or detach a thread that was previously joined or detached" id="BD-TRS-JOINDETACH" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Use locks to prevent race conditions when modifying bit fields" id="BD-TRS-BITLOCK" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Do not destroy another thread's mutex" id="BD-TRS-DSTRLOCK" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="BD-TRS" desc="Do not release a lock that has not been acquired" id="BD-TRS-REVLOCK" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CDD" desc="Avoid function duplication" id="CDD-DUPM" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CDD" desc="Avoid code duplication" id="CDD-DUPC" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="CDD" desc="Avoid duplication of #include directives" id="CDD-DUPI" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CDD" desc="Avoid string literal duplication" id="CDD-DUPS" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-API00" desc="The validity of parameters must be checked inside each function" id="CERT\_C-API00-a" sev="3">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Rule>

<Rule cat="CERT\_C-API01" desc="Avoid overflow when writing to a buffer" id="CERT\_C-API01-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-API01" desc="Avoid using unsafe string functions which may cause buffer overflows" id="CERT\_C-API01-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-API02" desc="Avoid using unsafe string functions which may cause buffer overflows" id="CERT\_C-API02-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-API02" desc="Don't use unsafe C functions that do write to range-unchecked buffers" id="CERT\_C-API02-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR01" desc="Do not call 'sizeof' on a pointer type" id="CERT\_C-ARR01-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR02" desc="Explicitly specify array bounds in array declarations with initializers" id="CERT\_C-ARR02-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR30" desc="Avoid accessing arrays out of bounds" id="CERT\_C-ARR30-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR32" desc="Ensure the size of the variable length array is in valid range" id="CERT\_C-ARR32-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR36" desc="Do not subtract two pointers that do not address elements of the same array" id="CERT\_C-ARR36-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR36" desc="Do not compare two unrelated pointers" id="CERT\_C-ARR36-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR37" desc="Pointer arithmetic shall not be applied to pointers that address variables of non-array type" id="CERT\_C-ARR37-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR38" desc="Avoid overflow when reading from a buffer" id="CERT\_C-ARR38-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR38" desc="Avoid overflow when writing to a buffer" id="CERT\_C-ARR38-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR38" desc="Avoid buffer overflow due to defining incorrect format limits" id="CERT\_C-ARR38-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR38" desc="Avoid overflow due to reading a not zero terminated string" id="CERT\_C-ARR38-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR39" desc="Avoid accessing arrays out of bounds" id="CERT\_C-ARR39-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR39" desc="Pointer arithmetic should not be used" id="CERT\_C-ARR39-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ARR39" desc="Do not add or subtract a scaled integer to a pointer" id="CERT\_C-ARR39-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON01" desc="Do not abandon unreleased locks" id="CERT\_C-CON01-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-CON02" desc="Do not use the volatile keyword" id="CERT\_C-CON02-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON05" desc="Do not use blocking functions while holding a lock" id="CERT\_C-CON05-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON30" desc="Ensure resources are freed" id="CERT\_C-CON30-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-CON31" desc="Do not destroy another thread's mutex" id="CERT\_C-CON31-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON31" desc="Do not use resources that have been freed" id="CERT\_C-CON31-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON31" desc="Do not free resources using invalid pointers" id="CERT\_C-CON31-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON32" desc="Use locks to prevent race conditions when modifying bit fields" id="CERT\_C-CON32-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON33" desc="Avoid using thread-unsafe functions" id="CERT\_C-CON33-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON34" desc="Declare objects shared between POSIX threads with appropriate storage durations" id="CERT\_C-CON34-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON35" desc="Avoid double locking" id="CERT\_C-CON35-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON36" desc="Wrap functions that can spuriously wake up in a loop" id="CERT\_C-CON36-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON37" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="CERT\_C-CON37-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON38" desc="Use the 'cnd\_signal()' function with a unique condition variable" id="CERT\_C-CON38-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON39" desc="Do not join or detach a thread that was previously joined or detached" id="CERT\_C-CON39-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON40" desc="Do not refer to an atomic variable twice in an expression" id="CERT\_C-CON40-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON41" desc="Wrap functions that can fail spuriously in a loop" id="CERT\_C-CON41-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-CON43" desc="Do not use global variable with different locks set" id="CERT\_C-CON43-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL00" desc="Declare parameters or local variable as const whenever possible" id="CERT\_C-DCL00-a" sev="3">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Rule>

<Rule cat="CERT\_C-DCL01" desc="Identifier declared in a local or function prototype scope shall not hide an identifier declared in a global or namespace scope" id="CERT\_C-DCL01-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL01" desc="Identifiers declared in an inner local scope should not hide identifiers declared in an outer local scope" id="CERT\_C-DCL01-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL02" desc="Use visually distinct identifiers" id="CERT\_C-DCL02-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL04" desc="Each variable should be declared in a separate declaration statement" id="CERT\_C-DCL04-a" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="CERT\_C-DCL05" desc="Declare a type of parameter as typedef to pointer to const if the pointer is not used to modify the addressed object" id="CERT\_C-DCL05-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL06" desc="&quot;#define&quot; or enum constants should be used instead of hard coded values whenever possible" id="CERT\_C-DCL06-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL10" desc="The number of format specifiers in the format string and the number of corresponding arguments in the invocation of a string formatting function should be equal" id="CERT\_C-DCL10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL11" desc="There should be no mismatch between the '%s' and '%c' format specifiers in the format string and their corresponding arguments in the invocation of a string formatting function" id="CERT\_C-DCL11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL11" desc="There should be no mismatch between the '%f' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="CERT\_C-DCL11-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL11" desc="There should be no mismatch between the '%i' and '%d' format specifiers in the string and their corresponding arguments in the invocation of a string formatting function" id="CERT\_C-DCL11-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL11" desc="There should be no mismatch between the '%u' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="CERT\_C-DCL11-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL11" desc="There should be no mismatch between the '%p' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="CERT\_C-DCL11-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL11" desc="The number of format specifiers in the format string and the number of corresponding arguments in the invocation of a string formatting function should be equal" id="CERT\_C-DCL11-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL12" desc="If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden" id="CERT\_C-DCL12-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL13" desc="A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object" id="CERT\_C-DCL13-a" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CERT\_C-DCL15" desc="Objects or functions with external linkage shall be declared in a header file" id="CERT\_C-DCL15-a" sev="3">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="CERT\_C-DCL16" desc="Use capital 'L' instead of lowercase 'l' to indicate long" id="CERT\_C-DCL16-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL18" desc="Octal and hexadecimal escape sequences shall be terminated" id="CERT\_C-DCL18-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL18" desc="Octal constants (other than zero) shall not be used" id="CERT\_C-DCL18-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL19" desc="Declare variables as locally as possible" id="CERT\_C-DCL19-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL20" desc="The number of arguments passed to a function shall match the number of parameters" id="CERT\_C-DCL20-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL22" desc="Avoid unused values" id="CERT\_C-DCL22-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL30" desc="The address of an object with automatic storage shall not be returned from a function" id="CERT\_C-DCL30-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL30" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="CERT\_C-DCL30-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL31" desc="All functions shall be declared before use" id="CERT\_C-DCL31-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL36" desc="Identifiers shall not simultaneously have both internal and external linkage in the same translation unit" id="CERT\_C-DCL36-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL37" desc="Do not #define or #undef identifiers with names which start with underscore" id="CERT\_C-DCL37-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL38" desc="The final member of a structure should not be an array of size '0' or '1'" id="CERT\_C-DCL38-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL39" desc="A pointer to a structure should not be passed to a function that can copy data to the user space" id="CERT\_C-DCL39-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL40" desc="All declarations of an object or function shall have compatible types" id="CERT\_C-DCL40-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL40" desc="If objects or functions are declared more than once their types shall be compatible" id="CERT\_C-DCL40-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-DCL41" desc="A switch statement shall only contain switch labels and switch clauses, and no other code" id="CERT\_C-DCL41-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV01" desc="Don't use unsafe C functions that do write to range-unchecked buffers" id="CERT\_C-ENV01-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV01" desc="Avoid using unsafe string functions which may cause buffer overflows" id="CERT\_C-ENV01-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV01" desc="Avoid overflow when writing to a buffer" id="CERT\_C-ENV01-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV02" desc="Usage of system properties (environment variables) should be restricted" id="CERT\_C-ENV02-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV30" desc="The pointers returned by the Standard Library functions 'localeconv', 'getenv', 'setlocale' or, 'strerror' shall only be used as if they have pointer to const-qualified type" id="CERT\_C-ENV30-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV31" desc="Do not rely on an environment pointer following an operation that may invalidate it" id="CERT\_C-ENV31-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV32" desc="Properly define exit handlers" id="CERT\_C-ENV32-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV33" desc="Do not call the 'system()' function from the 'stdlib.h' or 'cstdlib' library with an argument other than '0' (null pointer)" id="CERT\_C-ENV33-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ENV34" desc="Pointers returned by certain Standard Library functions should not be used following a subsequent call to the same or related function" id="CERT\_C-ENV34-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR01" desc="The error indicator 'errno' shall not be used" id="CERT\_C-ERR01-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR02" desc="The Standard Library input/output functions shall not be used" id="CERT\_C-ERR02-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CERT\_C-ERR04" desc="The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_C-ERR04-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR04" desc="The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_C-ERR04-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR04" desc="The 'quick\_exit()' and '\_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_C-ERR04-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR05" desc="The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_C-ERR05-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR05" desc="The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_C-ERR05-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR05" desc="The 'quick\_exit()' and '\_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_C-ERR05-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR06" desc="Do not use assertions" id="CERT\_C-ERR06-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR07" desc="The library functions atof, atoi and atol from library stdlib.h shall not be used" id="CERT\_C-ERR07-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CERT\_C-ERR07" desc="The Standard Library input/output functions shall not be used" id="CERT\_C-ERR07-b" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CERT\_C-ERR30" desc="Properly use errno value" id="CERT\_C-ERR30-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR30" desc="Provide error handling for file opening errors right next to the call to fopen" id="CERT\_C-ERR30-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR32" desc="Properly use errno value" id="CERT\_C-ERR32-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR33" desc="The value returned by a function having non-void return type shall be used" id="CERT\_C-ERR33-a" sev="1">

<Stats authTot="17;" authUrg="11;" total="17"/>

</Rule>

<Rule cat="CERT\_C-ERR33" desc="The value returned by a function having non-void return type shall be used" id="CERT\_C-ERR33-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR33" desc="Avoid null pointer dereferencing" id="CERT\_C-ERR33-c" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="CERT\_C-ERR33" desc="Always check the returned value of non-void function" id="CERT\_C-ERR33-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-ERR34" desc="The library functions atof, atoi and atol from library stdlib.h shall not be used" id="CERT\_C-ERR34-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CERT\_C-EXP00" desc="Use parenthesis to clarify expression order if operators with precedence lower than arithmetic are used" id="CERT\_C-EXP00-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP02" desc="The right-hand operand of a logical &amp;&amp; or || operator shall not contain side effects" id="CERT\_C-EXP02-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP05" desc="A cast shall not remove any 'const' or 'volatile' qualification from the type of a pointer or reference" id="CERT\_C-EXP05-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP08" desc="Pointer arithmetic should not be used" id="CERT\_C-EXP08-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP08" desc="Avoid accessing arrays out of bounds" id="CERT\_C-EXP08-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP10" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="CERT\_C-EXP10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP10" desc="Don't write code that depends on the order of evaluation of function arguments" id="CERT\_C-EXP10-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP10" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="CERT\_C-EXP10-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP10" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="CERT\_C-EXP10-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP12" desc="The value returned by a function having non-void return type shall be used" id="CERT\_C-EXP12-a" sev="3">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="CERT\_C-EXP12" desc="The value returned by a function having non-void return type shall be used" id="CERT\_C-EXP12-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP14" desc="Avoid mixing arithmetic of different precisions in the same expression" id="CERT\_C-EXP14-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP15" desc="Suspicious use of semicolon" id="CERT\_C-EXP15-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP16" desc="Function address should not be compared to zero" id="CERT\_C-EXP16-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP19" desc="The statement forming the body of a 'switch', 'while', 'do...while' or 'for' statement shall be a compound statement" id="CERT\_C-EXP19-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP20" desc="Avoid comparing values with TRUE macro/enum constant using equality operators (&quot;==&quot;, &quot;!=&quot;)" id="CERT\_C-EXP20-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP30" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="CERT\_C-EXP30-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP30" desc="Don't write code that depends on the order of evaluation of function arguments" id="CERT\_C-EXP30-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP30" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="CERT\_C-EXP30-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP30" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="CERT\_C-EXP30-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP32" desc="A cast shall not remove any 'const' or 'volatile' qualification from the type of a pointer or reference" id="CERT\_C-EXP32-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP33" desc="Avoid use before initialization" id="CERT\_C-EXP33-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP34" desc="Avoid null pointer dereferencing" id="CERT\_C-EXP34-a" sev="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-EXP35" desc="Do not access an array in the result of a function call" id="CERT\_C-EXP35-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP36" desc="A cast should not be performed between a pointer to object type and a different pointer to object type" id="CERT\_C-EXP36-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP37" desc="Identifiers shall be given for all of the parameters in a function prototype declaration" id="CERT\_C-EXP37-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP37" desc="Function types shall have named parameters" id="CERT\_C-EXP37-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP37" desc="Function types shall be in prototype form" id="CERT\_C-EXP37-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP37" desc="Functions shall always have visible prototype at the function call" id="CERT\_C-EXP37-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP39" desc="There shall be no implicit conversions from integral to floating type" id="CERT\_C-EXP39-a" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CERT\_C-EXP39" desc="A cast should not be performed between a pointer to object type and a different pointer to object type" id="CERT\_C-EXP39-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP39" desc="Avoid accessing arrays and pointers out of bounds" id="CERT\_C-EXP39-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP39" desc="Avoid buffer overflow from tainted data due to defining incorrect format limits" id="CERT\_C-EXP39-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP39" desc="Avoid buffer read overflow from tainted data" id="CERT\_C-EXP39-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP39" desc="Avoid buffer write overflow from tainted data" id="CERT\_C-EXP39-f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP40" desc="A cast shall not remove any 'const' or 'volatile' qualification from the type of a pointer or reference" id="CERT\_C-EXP40-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP42" desc="Don't memcpy or memcmp non-PODs" id="CERT\_C-EXP42-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP43" desc="The restrict type qualifier shall not be used" id="CERT\_C-EXP43-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP44" desc="Object designated by a volatile lvalue should not be accessed in the operand of the sizeof operator" id="CERT\_C-EXP44-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP44" desc="The function call that causes the side effect shall not be the operand of the sizeof operator" id="CERT\_C-EXP44-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP45" desc="Assignment operators shall not be used in conditions without brackets" id="CERT\_C-EXP45-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP45" desc="Assignment operators shall not be used in expressions that yield a Boolean value" id="CERT\_C-EXP45-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP46" desc="Expressions that are effectively Boolean should not be used as operands to operators other than (&amp;&amp;, ||, !, =, ==, !=, ?:)" id="CERT\_C-EXP46-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-EXP47" desc="Do not call va\_arg with an argument of the incorrect type" id="CERT\_C-EXP47-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO01" desc="Don't use chmod(), chown(), chgrp()" id="CERT\_C-FIO01-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO01" desc="Usage of functions prone to race is not allowed" id="CERT\_C-FIO01-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO21" desc="Usage of functions prone to race is not allowed" id="CERT\_C-FIO21-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO22" desc="Ensure resources are freed" id="CERT\_C-FIO22-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-FIO24" desc="Avoid race conditions while accessing files" id="CERT\_C-FIO24-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO30" desc="Avoid calling functions printf/wprintf with only one argument other than string constant" id="CERT\_C-FIO30-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO30" desc="Avoid using functions fprintf/fwprintf with only two parameters, when second parameter is a variable" id="CERT\_C-FIO30-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO30" desc="Never use unfiltered data from an untrusted user as the format parameter" id="CERT\_C-FIO30-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO32" desc="Protect against file name injection" id="CERT\_C-FIO32-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO34" desc="The macro EOF should be compared with the unmodified return value from the Standard Library function" id="CERT\_C-FIO34-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO37" desc="Avoid accessing arrays out of bounds" id="CERT\_C-FIO37-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO38" desc="A pointer to a FILE object shall not be dereferenced" id="CERT\_C-FIO38-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO39" desc="Do not alternately input and output from a stream without an intervening flush or positioning call" id="CERT\_C-FIO39-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO40" desc="Reset strings on fgets() or fgetws() failure" id="CERT\_C-FIO40-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO41" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="CERT\_C-FIO41-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO41" desc="Don't write code that depends on the order of evaluation of function arguments" id="CERT\_C-FIO41-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO41" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="CERT\_C-FIO41-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO41" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="CERT\_C-FIO41-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO41" desc="A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects" id="CERT\_C-FIO41-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO42" desc="Ensure resources are freed" id="CERT\_C-FIO42-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-FIO44" desc="Only use values for fsetpos() that are returned from fgetpos()" id="CERT\_C-FIO44-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO45" desc="Avoid race conditions while accessing files" id="CERT\_C-FIO45-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO46" desc="Do not use resources that have been freed" id="CERT\_C-FIO46-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO47" desc="There should be no mismatch between the '%s' and '%c' format specifiers in the format string and their corresponding arguments in the invocation of a string formatting function" id="CERT\_C-FIO47-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO47" desc="There should be no mismatch between the '%f' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="CERT\_C-FIO47-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO47" desc="There should be no mismatch between the '%i' and '%d' format specifiers in the string and their corresponding arguments in the invocation of a string formatting function" id="CERT\_C-FIO47-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO47" desc="There should be no mismatch between the '%u' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="CERT\_C-FIO47-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO47" desc="There should be no mismatch between the '%p' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="CERT\_C-FIO47-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FIO47" desc="The number of format specifiers in the format string and the number of corresponding arguments in the invocation of a string formatting function should be equal" id="CERT\_C-FIO47-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP00" desc="Floating-point expressions shall not be tested for equality or inequality" id="CERT\_C-FLP00-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP02" desc="Floating-point expressions shall not be tested for equality or inequality" id="CERT\_C-FLP02-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP03" desc="Avoid division by zero" id="CERT\_C-FLP03-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-FLP03" desc="Avoid implicit conversions from wider to narrower floating type" id="CERT\_C-FLP03-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP03" desc="Avoid implicit conversions from narrower to wider floating type" id="CERT\_C-FLP03-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP03" desc="Avoid implicit conversions of floating point numbers from wider to narrower floating type" id="CERT\_C-FLP03-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP06" desc="Implicit conversions from integral to floating type which may result in a loss of information shall not be used" id="CERT\_C-FLP06-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP06" desc="Implicit conversions from integral constant to floating type which may result in a loss of information shall not be used" id="CERT\_C-FLP06-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP30" desc="Do not use floating point variables as loop counters" id="CERT\_C-FLP30-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP32" desc="Validate values passed to library functions" id="CERT\_C-FLP32-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP34" desc="Avoid implicit conversions from wider to narrower floating type" id="CERT\_C-FLP34-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP34" desc="Avoid implicit conversions of floating point numbers from wider to narrower floating type" id="CERT\_C-FLP34-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP36" desc="Implicit conversions from integral to floating type which may result in a loss of information shall not be used" id="CERT\_C-FLP36-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP36" desc="Implicit conversions from integral constant to floating type which may result in a loss of information shall not be used" id="CERT\_C-FLP36-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-FLP37" desc="Do not use object representations to compare floating-point values" id="CERT\_C-FLP37-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT02" desc="Implicit conversions from wider to narrower integral type which may result in a loss of information shall not be used" id="CERT\_C-INT02-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT02" desc="Avoid mixing arithmetic of different precisions in the same expression" id="CERT\_C-INT02-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT04" desc="Protect against integer overflow/underflow from tainted data" id="CERT\_C-INT04-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT04" desc="Avoid buffer read overflow from tainted data" id="CERT\_C-INT04-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT04" desc="Avoid buffer write overflow from tainted data" id="CERT\_C-INT04-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT05" desc="Avoid using unsafe string functions that do not check bounds" id="CERT\_C-INT05-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CERT\_C-INT07" desc="The plain char type shall be used only for the storage and use of character values" id="CERT\_C-INT07-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT07" desc="signed and unsigned char type shall be used only for the storage and use of numeric values" id="CERT\_C-INT07-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT08" desc="Avoid integer overflows" id="CERT\_C-INT08-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT09" desc="In an enumerator list, the &quot;=&quot; construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised" id="CERT\_C-INT09-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT10" desc="Avoid accessing arrays out of bounds" id="CERT\_C-INT10-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT12" desc="Bit fields shall only be defined to be of type unsigned int or signed int" id="CERT\_C-INT12-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT13" desc="Bitwise operators shall only be applied to operands of unsigned underlying type" id="CERT\_C-INT13-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT15" desc="Use intmax\_t or uintmax\_t for formatted IO on programmer-defined integer types" id="CERT\_C-INT15-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT16" desc="Bitwise operators shall only be applied to operands of unsigned underlying type" id="CERT\_C-INT16-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT18" desc="Avoid possible integer overflow in expressions in which the result is assigned to a variable of a wider integer type" id="CERT\_C-INT18-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT18" desc="Avoid possible integer overflow in expressions in which the result is compared to an expression of a wider integer type" id="CERT\_C-INT18-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT18" desc="Avoid possible integer overflow in expressions in which the result is cast to a wider integer type" id="CERT\_C-INT18-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT30" desc="Avoid integer overflows" id="CERT\_C-INT30-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT30" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="CERT\_C-INT30-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT30" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="CERT\_C-INT30-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="An expression of essentially Boolean type should always be used where an operand is interpreted as a Boolean value" id="CERT\_C-INT31-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="An operand of essentially Boolean type should not be used where an operand is interpreted as a numeric value" id="CERT\_C-INT31-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="An operand of essentially character type should not be used where an operand is interpreted as a numeric value" id="CERT\_C-INT31-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="An operand of essentially enum type should not be used in an arithmetic operation" id="CERT\_C-INT31-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="Shift and bitwise operations should not be performed on operands of essentially signed or enum type" id="CERT\_C-INT31-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="An operand of essentially signed or enum type should not be used as the right hand operand to the bitwise shifting operator" id="CERT\_C-INT31-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="An operand of essentially unsigned type should not be used as the operand to the unary minus operator" id="CERT\_C-INT31-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="The value of an expression shall not be assigned to an object with a narrower essential type" id="CERT\_C-INT31-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="The value of an expression shall not be assigned to an object of a different essential type category" id="CERT\_C-INT31-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category" id="CERT\_C-INT31-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="The second and third operands of the ternary operator shall have the same essential type category" id="CERT\_C-INT31-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="The value of a composite expression shall not be assigned to an object with wider essential type" id="CERT\_C-INT31-l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type" id="CERT\_C-INT31-m" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="If a composite expression is used as one (second or third) operand of a conditional operator then the other operand shall not have wider essential type" id="CERT\_C-INT31-n" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT31" desc="Avoid integer overflows" id="CERT\_C-INT31-o" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT32" desc="Avoid integer overflows" id="CERT\_C-INT32-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT32" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="CERT\_C-INT32-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT32" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="CERT\_C-INT32-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT33" desc="Avoid division by zero" id="CERT\_C-INT33-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-INT34" desc="Avoid incorrect shift operations" id="CERT\_C-INT34-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT35" desc="Use correct integer precisions when checking the right hand operand of the shift operator" id="CERT\_C-INT35-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-INT36" desc="A conversion should not be performed between a pointer to object type and an integer type other than 'uintptr\_t' or 'intptr\_t'" id="CERT\_C-INT36-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM00" desc="Do not allocate memory and expect that someone else will deallocate it later" id="CERT\_C-MEM00-a" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="CERT\_C-MEM00" desc="Do not allocate memory and expect that someone else will deallocate it later" id="CERT\_C-MEM00-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM00" desc="Do not allocate memory and expect that someone else will deallocate it later" id="CERT\_C-MEM00-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM00" desc="Do not use resources that have been freed" id="CERT\_C-MEM00-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM00" desc="Ensure resources are freed" id="CERT\_C-MEM00-e" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="CERT\_C-MEM01" desc="Do not use resources that have been freed" id="CERT\_C-MEM01-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM01" desc="Always assign a new value to an expression that points to deallocated memory" id="CERT\_C-MEM01-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM01" desc="Always assign a new value to global or member variable that points to deallocated memory" id="CERT\_C-MEM01-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM01" desc="Always assign a new value to parameter or local variable that points to deallocated memory" id="CERT\_C-MEM01-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM02" desc="Assignment operator should have operands of compatible types" id="CERT\_C-MEM02-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM02" desc="Do not assign function return value to a variable of incompatible type" id="CERT\_C-MEM02-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM03" desc="Sensitive data should be cleared before being deallocated" id="CERT\_C-MEM03-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM04" desc="The validity of values passed to library functions shall be checked" id="CERT\_C-MEM04-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM05" desc="Do not use recursion" id="CERT\_C-MEM05-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM05" desc="Ensure the size of the variable length array is in valid range" id="CERT\_C-MEM05-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM07" desc="The validity of values passed to library functions shall be checked" id="CERT\_C-MEM07-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM12" desc="Ensure resources are freed" id="CERT\_C-MEM12-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-MEM30" desc="Do not use resources that have been freed" id="CERT\_C-MEM30-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM31" desc="Ensure resources are freed" id="CERT\_C-MEM31-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-MEM33" desc="Allocate structures containing a flexible array member dynamically" id="CERT\_C-MEM33-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM33" desc="Do not copy instances of structures containing a flexible array member" id="CERT\_C-MEM33-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM34" desc="Do not free resources using invalid pointers" id="CERT\_C-MEM34-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM35" desc="Do not use sizeof operator on pointer type to specify the size of the memory to be allocated via 'malloc', 'calloc' or 'realloc' function" id="CERT\_C-MEM35-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MEM36" desc="Do not modify the alignment of objects by calling realloc()" id="CERT\_C-MEM36-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC01" desc="All 'if...else-if' constructs shall be terminated with an 'else' clause" id="CERT\_C-MSC01-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC01" desc="The final clause of a switch statement shall be the default clause" id="CERT\_C-MSC01-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC04" desc="The character sequence /\* shall not be used within a C-style comment" id="CERT\_C-MSC04-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC04" desc="The character sequence // shall not be used within a C-style comment" id="CERT\_C-MSC04-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC04" desc="The character sequence /\* shall not be used within a C++-style comment" id="CERT\_C-MSC04-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC04" desc="Line-splicing shall not be used in // comments" id="CERT\_C-MSC04-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC07" desc="There shall be no unreachable code in &quot;else&quot; block" id="CERT\_C-MSC07-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC07" desc="There shall be no unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="CERT\_C-MSC07-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC07" desc="There shall be no unreachable code in &quot;if/else/while/for&quot; block" id="CERT\_C-MSC07-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC07" desc="There shall be no unreachable code in switch statement" id="CERT\_C-MSC07-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC07" desc="There shall be no unreachable code in 'for' loop" id="CERT\_C-MSC07-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC07" desc="There shall be no unreachable code after 'if' or 'switch' statement" id="CERT\_C-MSC07-f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC07" desc="There shall be no unreachable code after &quot;if&quot; or &quot;switch&quot; statement inside while/for/do...while loop" id="CERT\_C-MSC07-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC07" desc="Avoid switch with unreachable branches" id="CERT\_C-MSC07-h" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC09" desc="Only use characters defined in ISO C standard" id="CERT\_C-MSC09-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC11" desc="Assert liberally to document internal assumptions and invariants" id="CERT\_C-MSC11-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC12" desc="There shall be no unreachable code in &quot;else&quot; block" id="CERT\_C-MSC12-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC12" desc="There shall be no unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="CERT\_C-MSC12-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC12" desc="There shall be no unreachable code in &quot;if/else/while/for&quot; block" id="CERT\_C-MSC12-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC12" desc="There shall be no unreachable code in switch statement" id="CERT\_C-MSC12-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC12" desc="There shall be no unreachable code in 'for' loop" id="CERT\_C-MSC12-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC12" desc="There shall be no unreachable code after 'if' or 'switch' statement" id="CERT\_C-MSC12-f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC12" desc="There shall be no unreachable code after &quot;if&quot; or &quot;switch&quot; statement inside while/for/do...while loop" id="CERT\_C-MSC12-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC12" desc="Avoid switch with unreachable branches" id="CERT\_C-MSC12-h" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC13" desc="Avoid unnecessary local variables" id="CERT\_C-MSC13-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC14" desc="Evaluation of constant unsigned integer expressions should not lead to wrap-around" id="CERT\_C-MSC14-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC15" desc="Evaluation of constant unsigned integer expressions should not lead to wrap-around" id="CERT\_C-MSC15-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC17" desc="Missing break statement between cases in a switch statement" id="CERT\_C-MSC17-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC19" desc="Avoid accessing arrays out of bounds" id="CERT\_C-MSC19-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC19" desc="Avoid null pointer dereferencing" id="CERT\_C-MSC19-b" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_C-MSC22" desc="The setjmp macro and the longjmp function shall not be used" id="CERT\_C-MSC22-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC24" desc="The library functions atof, atoi and atol from library stdlib.h shall not be used" id="CERT\_C-MSC24-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CERT\_C-MSC24" desc="The 'getenv()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_C-MSC24-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC24" desc="Avoid using unsafe string functions which may cause buffer overflows" id="CERT\_C-MSC24-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC24" desc="Don't use unsafe C functions that do write to range-unchecked buffers" id="CERT\_C-MSC24-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC30" desc="Do not use the rand() function for generating pseudorandom numbers" id="CERT\_C-MSC30-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC32" desc="Properly seed pseudorandom number generators" id="CERT\_C-MSC32-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC33" desc="Avoid functions which use time from standard C library" id="CERT\_C-MSC33-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC37" desc="All exit paths from a function, except main(), with non-void return type shall have an explicit return statement with an expression" id="CERT\_C-MSC37-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC38" desc="A function-like macro shall not be invoked without all of its arguments" id="CERT\_C-MSC38-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC39" desc="Use macros for variable arguments correctly" id="CERT\_C-MSC39-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC40" desc="An inline definition of a function with external linkage shall not contain definitions and uses of static objects" id="CERT\_C-MSC40-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-MSC41" desc="Do not hard code string literals" id="CERT\_C-MSC41-a" sev="1">

<Stats authTot="3;" authUrg="2;" total="3"/>

</Rule>

<Rule cat="CERT\_C-POS30" desc="Avoid overflow due to reading a not zero terminated string" id="CERT\_C-POS30-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS30" desc="The values returned by functions 'read' and 'readlink' shall be used" id="CERT\_C-POS30-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS30" desc="Use of possibly not null-terminated string with functions expecting null-terminated string" id="CERT\_C-POS30-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS33" desc="Avoid using the 'vfork()' function" id="CERT\_C-POS33-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS34" desc="Usage of system properties (environment variables) should be restricted" id="CERT\_C-POS34-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS34" desc="Do not call putenv() with a pointer to an automatic variable as the argument" id="CERT\_C-POS34-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS35" desc="Avoid race conditions while checking for the existence of a symbolic link" id="CERT\_C-POS35-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS36" desc="Observe correct revocation order while relinquishing privileges" id="CERT\_C-POS36-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS37" desc="Ensure that privilege relinquishment is successful" id="CERT\_C-POS37-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS38" desc="Avoid race conditions when using fork and file descriptors" id="CERT\_C-POS38-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS39" desc="Use the correct byte ordering when transferring data between systems" id="CERT\_C-POS39-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS44" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="CERT\_C-POS44-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS47" desc="The function 'pthread\_setcanceltype()' should not be called with 'PTHREAD\_CANCEL\_ASYNCHRONOUS' argument" id="CERT\_C-POS47-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS48" desc="Do not destroy another thread's mutex" id="CERT\_C-POS48-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS48" desc="Do not release a lock that has not been acquired" id="CERT\_C-POS48-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS49" desc="Use locks to prevent race conditions when modifying bit fields" id="CERT\_C-POS49-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS50" desc="Declare objects shared between POSIX threads with appropriate storage durations" id="CERT\_C-POS50-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS51" desc="Do not acquire locks in different order" id="CERT\_C-POS51-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS52" desc="Do not use blocking functions while holding a lock" id="CERT\_C-POS52-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS53" desc="Do not use more than one mutex for concurrent waiting operations on a condition variable" id="CERT\_C-POS53-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS54" desc="The value returned by a function having non-void return type shall be used" id="CERT\_C-POS54-a" sev="1">

<Stats authTot="17;" authUrg="11;" total="17"/>

</Rule>

<Rule cat="CERT\_C-POS54" desc="The value returned by a function having non-void return type shall be used" id="CERT\_C-POS54-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-POS54" desc="Avoid null pointer dereferencing" id="CERT\_C-POS54-c" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="CERT\_C-PRE00" desc="A function should be used in preference to a function-like macro" id="CERT\_C-PRE00-a" sev="3">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="CERT\_C-PRE01" desc="In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##" id="CERT\_C-PRE01-a" sev="1">

<Stats authTot="8;" authUrg="2;" total="8"/>

</Rule>

<Rule cat="CERT\_C-PRE02" desc="Enclose in parentheses whole definition of a function-like macro" id="CERT\_C-PRE02-a" sev="1">

<Stats authTot="8;" authUrg="6;" total="8"/>

</Rule>

<Rule cat="CERT\_C-PRE06" desc="Use multiple include guards" id="CERT\_C-PRE06-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-PRE07" desc="Trigraphs shall not be used" id="CERT\_C-PRE07-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-PRE30" desc="Avoid token concatenation that may produce universal character names" id="CERT\_C-PRE30-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-PRE31" desc="Assertions should not contain assignments, increment, or decrement operators" id="CERT\_C-PRE31-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-PRE31" desc="Assertions should not contain function calls nor function-like macro calls" id="CERT\_C-PRE31-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-PRE31" desc="Avoid side effects in arguments to unsafe macros" id="CERT\_C-PRE31-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-PRE32" desc="Arguments to a function-like macro shall not contain tokens that look like preprocessing directives" id="CERT\_C-PRE32-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-SIG00" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="CERT\_C-SIG00-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-SIG01" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="CERT\_C-SIG01-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-SIG02" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="CERT\_C-SIG02-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-SIG30" desc="Properly define signal handlers" id="CERT\_C-SIG30-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-SIG31" desc="Properly define signal handlers" id="CERT\_C-SIG31-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-SIG34" desc="Properly define signal handlers" id="CERT\_C-SIG34-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-SIG35" desc="Do not return from a computational exception signal handler" id="CERT\_C-SIG35-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR00" desc="The plain char type shall be used only for the storage and use of character values" id="CERT\_C-STR00-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR02" desc="Protect against command injection" id="CERT\_C-STR02-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR02" desc="Protect against file name injection" id="CERT\_C-STR02-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR02" desc="Protect against SQL injection" id="CERT\_C-STR02-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR03" desc="Avoid overflow due to reading a not zero terminated string" id="CERT\_C-STR03-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR04" desc="The plain char type shall be used only for the storage and use of character values" id="CERT\_C-STR04-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR05" desc="A string literal shall not be modified" id="CERT\_C-STR05-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR07" desc="Avoid using unsafe string functions that do not check bounds" id="CERT\_C-STR07-a" sev="1">

<Stats authTot="2;" authUrg="1;" total="2"/>

</Rule>

<Rule cat="CERT\_C-STR09" desc="Expressions with type (plain) char and wchar\_t shall not be used as operands to built-in operators other than =, ==, != and the unary &amp; operator" id="CERT\_C-STR09-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR10" desc="Narrow and wide string literals shall not be concatenated" id="CERT\_C-STR10-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR11" desc="Do not specify the bound of a character array initialized with a string literal" id="CERT\_C-STR11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR30" desc="A string literal shall not be modified" id="CERT\_C-STR30-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR30" desc="Do not modify string literals" id="CERT\_C-STR30-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR31" desc="Avoid accessing arrays out of bounds" id="CERT\_C-STR31-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR31" desc="Avoid overflow when writing to a buffer" id="CERT\_C-STR31-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR31" desc="Prevent buffer overflows from tainted data" id="CERT\_C-STR31-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR31" desc="Avoid buffer write overflow from tainted data" id="CERT\_C-STR31-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR31" desc="Avoid using unsafe string functions which may cause buffer overflows" id="CERT\_C-STR31-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR32" desc="Avoid overflow due to reading a not zero terminated string" id="CERT\_C-STR32-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR34" desc="Cast characters to unsigned char before assignment to larger integer sizes" id="CERT\_C-STR34-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR34" desc="An expressions of the 'signed char' type should not be used as an array index" id="CERT\_C-STR34-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR34" desc="Cast characters to unsigned char before converting to larger integer sizes" id="CERT\_C-STR34-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR37" desc="Do not pass incorrect values to ctype.h library functions" id="CERT\_C-STR37-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-STR38" desc="Do not confuse narrow and wide character strings and functions" id="CERT\_C-STR38-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-WIN00" desc="Use care to ensure that LoadLibrary() will load the correct library" id="CERT\_C-WIN00-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_C-WIN30" desc="Ensure resources are freed" id="CERT\_C-WIN30-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_CPP-CON50" desc="Do not destroy another thread's mutex" id="CERT\_CPP-CON50-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CON51" desc="Do not call lock() directly on a mutex" id="CERT\_CPP-CON51-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CON52" desc="Use locks to prevent race conditions when modifying bit fields" id="CERT\_CPP-CON52-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CON53" desc="Do not acquire locks in different order" id="CERT\_CPP-CON53-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CON54" desc="Wrap functions that can spuriously wake up in a loop" id="CERT\_CPP-CON54-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CON55" desc="Do not use the 'notify\_one()' function when multiple threads are waiting on the same condition variable" id="CERT\_CPP-CON55-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CON56" desc="Avoid double locking" id="CERT\_CPP-CON56-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR50" desc="Guarantee that container indices are within the valid range" id="CERT\_CPP-CTR50-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR51" desc="Do not modify container while iterating over it" id="CERT\_CPP-CTR51-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR52" desc="Do not pass empty container iterators to std algorithms as destinations" id="CERT\_CPP-CTR52-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR53" desc="Do not use an iterator range that isn't really a range" id="CERT\_CPP-CTR53-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR53" desc="Do not compare iterators from different containers" id="CERT\_CPP-CTR53-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR54" desc="Do not compare iterators from different containers" id="CERT\_CPP-CTR54-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR54" desc="Do not compare two unrelated pointers" id="CERT\_CPP-CTR54-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR54" desc="Do not subtract two pointers that do not address elements of the same array" id="CERT\_CPP-CTR54-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR55" desc="Do not add or subtract a constant with a value greater than one from an iterator" id="CERT\_CPP-CTR55-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR56" desc="Don't treat arrays polymorphically" id="CERT\_CPP-CTR56-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR56" desc="A pointer to an array of derived class objects should not be converted to a base class pointer" id="CERT\_CPP-CTR56-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR56" desc="Do not treat arrays polymorphically" id="CERT\_CPP-CTR56-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR57" desc="For associative containers never use comparison function returning true for equal values" id="CERT\_CPP-CTR57-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-CTR58" desc="Make predicates const pure functions" id="CERT\_CPP-CTR58-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL50" desc="Functions shall not be defined with a variable number of arguments" id="CERT\_CPP-DCL50-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL51" desc="Do not #define or #undef identifiers with names which start with underscore" id="CERT\_CPP-DCL51-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL51" desc="Do not redefine reserved words" id="CERT\_CPP-DCL51-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL51" desc="Do not #define nor #undef identifier 'defined'" id="CERT\_CPP-DCL51-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL51" desc="The names of standard library macros, objects and functions shall not be reused" id="CERT\_CPP-DCL51-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL51" desc="The names of standard library macros, objects and functions shall not be reused (C90)" id="CERT\_CPP-DCL51-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL51" desc="The names of standard library macros, objects and functions shall not be reused (C99)" id="CERT\_CPP-DCL51-f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL52" desc="Never qualify a reference type with 'const' or 'volatile'" id="CERT\_CPP-DCL52-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL53" desc="Parameter names in function declarations should not be enclosed in parentheses" id="CERT\_CPP-DCL53-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL53" desc="Local variable names in variable declarations should not be enclosed in parentheses" id="CERT\_CPP-DCL53-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL53" desc="Avoid function declarations that are syntactically ambiguous" id="CERT\_CPP-DCL53-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL54" desc="Always provide new and delete together" id="CERT\_CPP-DCL54-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL55" desc="A pointer to a structure should not be passed to a function that can copy data to the user space" id="CERT\_CPP-DCL55-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL56" desc="Avoid initialization order problems across translation units by replacing non-local static objects with local static objects" id="CERT\_CPP-DCL56-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL57" desc="Never allow an exception to be thrown from a destructor, deallocation, and swap" id="CERT\_CPP-DCL57-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL57" desc="Always catch exceptions" id="CERT\_CPP-DCL57-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL58" desc="Do not modify the standard namespaces 'std' and 'posix'" id="CERT\_CPP-DCL58-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL59" desc="There shall be no unnamed namespaces in header files" id="CERT\_CPP-DCL59-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-DCL60" desc="A class, union or enum name (including qualification, if any) shall be a unique identifier" id="CERT\_CPP-DCL60-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="The execution of a function registered with 'std::atexit()' or 'std::at\_quick\_exit()' should not exit via an exception" id="CERT\_CPP-ERR50-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Never allow an exception to be thrown from a destructor, deallocation, and swap" id="CERT\_CPP-ERR50-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Do not throw from within destructor" id="CERT\_CPP-ERR50-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="There should be at least one exception handler to catch all otherwise unhandled exceptions" id="CERT\_CPP-ERR50-d" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="An empty throw (throw;) shall only be used in the compound-statement of a catch handler" id="CERT\_CPP-ERR50-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Exceptions shall be raised only after start-up and before termination of the program" id="CERT\_CPP-ERR50-f" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="CERT\_CPP-ERR50-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Where a function's declaration includes an exception-specification, the function shall only be capable of throwing exceptions of the indicated type(s)" id="CERT\_CPP-ERR50-h" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Function called in global or namespace scope shall not throw unhandled exceptions" id="CERT\_CPP-ERR50-i" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Always catch exceptions" id="CERT\_CPP-ERR50-j" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Properly define exit handlers" id="CERT\_CPP-ERR50-k" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_CPP-ERR50-l" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="Avoid throwing exceptions from functions that are declared not to throw" id="CERT\_CPP-ERR50-m" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR50" desc="The 'quick\_exit()' and '\_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used" id="CERT\_CPP-ERR50-n" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR51" desc="Always catch exceptions" id="CERT\_CPP-ERR51-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR51" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="CERT\_CPP-ERR51-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR52" desc="The setjmp macro and the longjmp function shall not be used" id="CERT\_CPP-ERR52-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR52" desc="The standard header file &lt;setjmp.h> shall not be used" id="CERT\_CPP-ERR52-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR53" desc="Handlers of a function-try-block implementation of a class constructor or destructor shall not reference nonstatic members from this class or its bases" id="CERT\_CPP-ERR53-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR54" desc="Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class" id="CERT\_CPP-ERR54-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR55" desc="Where a function's declaration includes an exception-specification, the function shall only be capable of throwing exceptions of the indicated type(s)" id="CERT\_CPP-ERR55-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR56" desc="Always catch exceptions" id="CERT\_CPP-ERR56-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR56" desc="Do not leave 'catch' blocks empty" id="CERT\_CPP-ERR56-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR57" desc="Ensure resources are freed" id="CERT\_CPP-ERR57-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_CPP-ERR58" desc="Exceptions shall be raised only after start-up and before termination of the program" id="CERT\_CPP-ERR58-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_CPP-ERR59" desc="Do not throw an exception across execution boundaries" id="CERT\_CPP-ERR59-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR60" desc="Exception objects must be nothrow copy constructible" id="CERT\_CPP-ERR60-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR60" desc="An explicitly declared copy constructor for a class that inherits from 'std::exception' should have a non-throwing exception specification" id="CERT\_CPP-ERR60-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR61" desc="A class type exception shall always be caught by reference" id="CERT\_CPP-ERR61-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR61" desc="Throw by value, catch by reference" id="CERT\_CPP-ERR61-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-ERR62" desc="The library functions atof, atoi and atol from library stdlib.h shall not be used" id="CERT\_CPP-ERR62-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CERT\_CPP-EXP50" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="CERT\_CPP-EXP50-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP50" desc="Don't write code that depends on the order of evaluation of function arguments" id="CERT\_CPP-EXP50-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP50" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="CERT\_CPP-EXP50-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP50" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="CERT\_CPP-EXP50-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP50" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="CERT\_CPP-EXP50-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP50" desc="Don't write code that depends on the order of evaluation of function calls" id="CERT\_CPP-EXP50-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP51" desc="Do not treat arrays polymorphically" id="CERT\_CPP-EXP51-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP52" desc="The operand of the sizeof operator shall not contain any expression which has side effects" id="CERT\_CPP-EXP52-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP52" desc="Object designated by a volatile lvalue should not be accessed in the operand of the sizeof operator" id="CERT\_CPP-EXP52-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP52" desc="The function call that causes the side effect shall not be the operand of the sizeof operator" id="CERT\_CPP-EXP52-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP52" desc="The operand of the 'typeid' operator shall not contain any expression that has side effects" id="CERT\_CPP-EXP52-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP52" desc="The operand of the 'typeid' operator shall not contain a function call that causes side effects" id="CERT\_CPP-EXP52-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP53" desc="Avoid use before initialization" id="CERT\_CPP-EXP53-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP54" desc="Do not use resources that have been freed" id="CERT\_CPP-EXP54-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP54" desc="The address of an object with automatic storage shall not be returned from a function" id="CERT\_CPP-EXP54-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP54" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="CERT\_CPP-EXP54-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP55" desc="A cast shall not remove any 'const' or 'volatile' qualification from the type of a pointer or reference" id="CERT\_CPP-EXP55-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP56" desc="Do not call a function with a mismatched language linkage" id="CERT\_CPP-EXP56-a" sev="3">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="CERT\_CPP-EXP57" desc="Do not delete objects with incomplete class at the point of deletion" id="CERT\_CPP-EXP57-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP57" desc="Conversions shall not be performed between a pointer to an incomplete type and any other type" id="CERT\_CPP-EXP57-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP58" desc="Use macros for variable arguments correctly" id="CERT\_CPP-EXP58-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP59" desc="Use offsetof() on valid types and members" id="CERT\_CPP-EXP59-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP60" desc="Do not pass a nonstandard-layout type object across execution boundaries" id="CERT\_CPP-EXP60-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP61" desc="Never return lambdas that capture local objects by reference" id="CERT\_CPP-EXP61-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP61" desc="Never capture local objects from an outer lambda by reference" id="CERT\_CPP-EXP61-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP61" desc="The lambda that captures local objects by reference should not be assigned to the variable with a greater lifetime" id="CERT\_CPP-EXP61-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP62" desc="Do not compare objects of a class that may contain padding bits with C standard library functions" id="CERT\_CPP-EXP62-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-EXP63" desc="Do not rely on the value of a moved-from object" id="CERT\_CPP-EXP63-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-FIO50" desc="Do not alternately input and output from a stream without an intervening flush or positioning call" id="CERT\_CPP-FIO50-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-FIO51" desc="Ensure resources are freed" id="CERT\_CPP-FIO51-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_CPP-INT50" desc="An expression with enum underlying type shall only have values corresponding to the enumerators of the enumeration" id="CERT\_CPP-INT50-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM50" desc="Do not use resources that have been freed" id="CERT\_CPP-MEM50-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM51" desc="Use the same form in corresponding calls to new/malloc and delete/free" id="CERT\_CPP-MEM51-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM51" desc="Always provide empty brackets ([]) for delete when deallocating arrays" id="CERT\_CPP-MEM51-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM51" desc="Both copy constructor and copy assignment operator should be declared for classes with a nontrivial destructor" id="CERT\_CPP-MEM51-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM51" desc="Properly deallocate dynamically allocated resources" id="CERT\_CPP-MEM51-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM52" desc="Check the return value of new" id="CERT\_CPP-MEM52-a" sev="1">

<Stats authTot="2;" authUrg="1;" total="2"/>

</Rule>

<Rule cat="CERT\_CPP-MEM52" desc="Do not allocate resources in function argument list because the order of evaluation of a function's parameters is undefined" id="CERT\_CPP-MEM52-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM53" desc="Do not invoke malloc/realloc for objects having constructors" id="CERT\_CPP-MEM53-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM54" desc="Do not pass a pointer that has insufficient storage capacity or that is not suitably aligned for the object being constructed to placement 'new'" id="CERT\_CPP-MEM54-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM54" desc="An overhead should be used when an array of objects is passed to the placement 'new' allocation function" id="CERT\_CPP-MEM54-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM55" desc="The user defined 'new' operator should throw the 'std::bad\_alloc' exception when the allocation fails" id="CERT\_CPP-MEM55-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM56" desc="Do not store an already-owned pointer value in an unrelated smart pointer" id="CERT\_CPP-MEM56-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MEM57" desc="Avoid using the default operator 'new' for over-aligned types" id="CERT\_CPP-MEM57-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MSC50" desc="Do not use the rand() function for generating pseudorandom numbers" id="CERT\_CPP-MSC50-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MSC51" desc="Properly seed pseudorandom number generators" id="CERT\_CPP-MSC51-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MSC52" desc="All exit paths from a function, except main(), with non-void return type shall have an explicit return statement with an expression" id="CERT\_CPP-MSC52-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MSC53" desc="Never return from functions that should not return" id="CERT\_CPP-MSC53-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-MSC54" desc="Properly define signal handlers" id="CERT\_CPP-MSC54-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP50" desc="Avoid calling virtual functions from constructors" id="CERT\_CPP-OOP50-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP50" desc="Avoid calling virtual functions from destructors" id="CERT\_CPP-OOP50-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP50" desc="Do not use dynamic type of an object under construction" id="CERT\_CPP-OOP50-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP50" desc="Do not use dynamic type of an object under destruction" id="CERT\_CPP-OOP50-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP51" desc="Avoid slicing function arguments / return value" id="CERT\_CPP-OOP51-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP52" desc="Define a virtual destructor in classes used as base classes which have virtual functions" id="CERT\_CPP-OOP52-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_CPP-OOP53" desc="List members in an initialization list in the order in which they are declared" id="CERT\_CPP-OOP53-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP54" desc="Check for assignment to self in operator=" id="CERT\_CPP-OOP54-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP55" desc="A cast shall not convert a pointer to a function to any other pointer type, including a pointer to function type" id="CERT\_CPP-OOP55-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP56" desc="Properly define terminate handlers" id="CERT\_CPP-OOP56-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP56" desc="Properly define unexpected handlers" id="CERT\_CPP-OOP56-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP56" desc="Properly define new handlers" id="CERT\_CPP-OOP56-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP57" desc="Do not initialize objects with a non-trivial class type using C standard library functions" id="CERT\_CPP-OOP57-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP57" desc="Do not compare objects of nonstandard-layout class type with C standard library functions" id="CERT\_CPP-OOP57-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-OOP58" desc="Copy operations must not mutate the source object" id="CERT\_CPP-OOP58-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-STR50" desc="Avoid overflow due to reading a not zero terminated string" id="CERT\_CPP-STR50-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-STR50" desc="Avoid overflow when writing to a buffer" id="CERT\_CPP-STR50-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-STR50" desc="Prevent buffer overflows from tainted data" id="CERT\_CPP-STR50-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-STR50" desc="Avoid buffer write overflow from tainted data" id="CERT\_CPP-STR50-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-STR50" desc="Do not use the 'char' buffer to store input from 'std::cin'" id="CERT\_CPP-STR50-g" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-STR51" desc="Avoid null pointer dereferencing" id="CERT\_CPP-STR51-a" sev="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CERT\_CPP-STR52" desc="Use valid references, pointers, and iterators to reference elements of a basic\_string" id="CERT\_CPP-STR52-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CERT\_CPP-STR53" desc="Guarantee that container indices are within the valid range" id="CERT\_CPP-STR53-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Local variable or parameter names and class member variable or parent class/struct member variable names shall differ by more than a single character" id="CODSTA-22" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid internal or external name conflict with a C++ reserved word" id="CODSTA-39" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Local variables should not use the same names as member variables" id="CODSTA-44" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Parameters should not use the same names as member variables" id="CODSTA-45" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use break in for loops" id="CODSTA-08" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Global functions shall not declare parameters with array type" id="CODSTA-142" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA" desc="Do not declare the size of an array when the array is passed into a function as a parameter" id="CODSTA-15" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not declare the size of an array when the array is initialized" id="CODSTA-16" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The declaration of an array parameter shall not contain the 'static' keyword between the [ ]" id="CODSTA-160" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A pointer to a FILE object shall not be dereferenced" id="CODSTA-166\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A pointer to a FILE object shall not be dereferenced by a library function" id="CODSTA-166\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The 'sizeof' operator shall not have an operand which is a function parameter declared as &quot;array of type&quot;" id="CODSTA-182" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The pointers returned by the Standard Library functions 'localeconv', 'getenv', 'setlocale' or, 'strerror' shall only be used as if they have pointer to const-qualified type" id="CODSTA-185\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not specify the bound of a character array initialized with a string literal" id="CODSTA-197" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="EOS should be used to terminate a string rather than NULL" id="CODSTA-20" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An inline definition of a function with external linkage shall not contain definitions and uses of static objects" id="CODSTA-202" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Literal suffixes shall use uppercase rather than lowercase letters" id="CODSTA-51" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Enumeration types shall be used instead of integer types (and constants) as case labels" id="CODSTA-55" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Array elements shall be accessed by the array operator [ ]" id="CODSTA-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not define constants via #define" id="CODSTA-03" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA" desc="Pointers to pointers should be avoided whenever possible" id="CODSTA-05" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid using the '?:' operator" id="CODSTA-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="If a function has no parameters, use ( ) instead of ( void )" id="CODSTA-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not cast pointers to functions to pointers to primitive types" id="CODSTA-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Storage type modifiers shall be associated with the type, not the variable or the function" id="CODSTA-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Signed and unsigned values shall not be mixed in arithmetic operations" id="CODSTA-100" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Signed and unsigned values shall not be mixed in second and third operand of conditional operator" id="CODSTA-101" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A single operation with side-effect shall only be used in the proper context" id="CODSTA-102" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A call of function with side-effect shall only be used in the proper context" id="CODSTA-103" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The operands of a logical &amp;&amp; or || shall be parenthesized if the operands contain binary operators" id="CODSTA-104" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The library functions bsearch and qsort of &lt;stdlib.h> shall not be used" id="CODSTA-107" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The standard header file &lt;tgmath.h> shall not be used" id="CODSTA-108" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The facilities that are specified as being provided by &lt;tgmath.h> should not be used" id="CODSTA-108\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The Standard Library input/output functions shall not be used" id="CODSTA-110" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA" desc="The union keyword should not be used" id="CODSTA-111" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Variable-length array types shall not be used" id="CODSTA-112" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Flexible array members shall not be declared" id="CODSTA-113" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Assembly language shall be encapsulated and isolated in C/C++ functions" id="CODSTA-114" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A 'default' label, if it exists, shall appear as either the first or the last switch label of a switch statement" id="CODSTA-116" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Octal and hexadecimal escape sequences shall be terminated" id="CODSTA-117" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A declaration shall be visible when an object or function with external linkage is defined" id="CODSTA-118" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A 'default' label shall have a statement or a comment before terminating 'break'" id="CODSTA-119" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid using shift operations instead of arithmetic operations" id="CODSTA-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An inline function shall be declared with the static storage class" id="CODSTA-120" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The restrict type qualifier shall not be used" id="CODSTA-121" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The value returned by a function having non-void return type shall be used" id="CODSTA-122\_a" sev="3">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="CODSTA" desc="The value returned by a function having non-void return type shall be used" id="CODSTA-122\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Conversions shall not be performed between a pointer to a function and any other type than pointer to function" id="CODSTA-124\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Conversions shall not be performed between non compatible pointer to a function types" id="CODSTA-124\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Conversions shall not be performed between a pointer to an incomplete type and any other type" id="CODSTA-125" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A cast shall not be performed between a pointer to object type and a pointer to a different object type" id="CODSTA-126" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A cast shall not be performed between pointer to void and an arithmetic type" id="CODSTA-129\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An implicit conversion shall not be performed between pointer to void and an arithmetic type" id="CODSTA-129\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc=">, >=, &lt;, &lt;= shall not be applied to pointer types except where they point to the same array" id="CODSTA-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A cast shall not be performed between pointer to object and a non-integer arithmetic type" id="CODSTA-130" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Use NULL instead of literal zero (0) as the null-pointer-constant" id="CODSTA-131" sev="3">

<Stats authTot="19;" authUrg="0;" total="19"/>

</Rule>

<Rule cat="CODSTA" desc="A macro shall not be defined with the same name as a keyword in C90" id="CODSTA-133\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A macro shall not be defined with the same name as a keyword in C99" id="CODSTA-133\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Type casting from any type to or from pointers shall not be used" id="CODSTA-135" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The identifiers va\_list, va\_arg, va\_start, va\_end, va\_copy should not be used" id="CODSTA-136\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The identifiers va\_list, va\_arg, va\_start, va\_end should not be used" id="CODSTA-136\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The function call shall not be the operand of the sizeof operator" id="CODSTA-137" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The result of a built-in assignment operator should not be used" id="CODSTA-138" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Pointer subtraction shall only be applied to pointers that address elements of the same array" id="CODSTA-13\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Never convert consts to non-consts" id="CODSTA-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Initializer lists shall not contain persistent side effects" id="CODSTA-141\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Functions which are called with array argument shall not be declared with pointer parameter" id="CODSTA-142\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Suspicious use of 'strcpy' without checking size of source buffer" id="CODSTA-143" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The values returned by functions 'read' and 'readlink' shall be used" id="CODSTA-144" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Use of possibly not null-terminated string with functions expecting null-terminated string" id="CODSTA-145" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A value returned from a stdio character-oriented I/O function should not be implicitly converted to a type smaller than 'int'" id="CODSTA-146" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Pointer should not be compared with NULL using relational operators &lt;, >, >=, &lt;=" id="CODSTA-147" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use string literals as operands of equality or relational operators" id="CODSTA-148" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Missing break statement between cases in a switch statement" id="CODSTA-149" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Assertions should not contain assignments, increment, or decrement operators" id="CODSTA-150" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid side effects in arguments to unsafe macros" id="CODSTA-150\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The value of a composite expression shall not be cast to a different essential type category or a wider essential type" id="CODSTA-152" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An external object or function shall not have more than one non-defining declaration in translation unit" id="CODSTA-154" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Use typedefs from stdint.h instead of declaring your own in C99 code" id="CODSTA-155" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type" id="CODSTA-156\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="If a composite expression is used as one (second or third) operand of a conditional operator then the other operand shall not have wider essential type" id="CODSTA-156\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The value of a composite expression shall not be assigned to an object with wider essential type" id="CODSTA-157" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Function types shall have named parameters" id="CODSTA-158" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Function types shall be in prototype form" id="CODSTA-159" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An expression of essentially Boolean type should always be used where an operand is interpreted as a Boolean value" id="CODSTA-161\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An operand of essentially Boolean type should not be used where an operand is interpreted as a numeric value" id="CODSTA-161\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An operand of essentially character type should not be used where an operand is interpreted as a numeric value" id="CODSTA-161\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An operand of essentially enum type should not be used in an arithmetic operation" id="CODSTA-161\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Shift and bitwise operations should not be performed on operands of essentially signed or enum type" id="CODSTA-161\_e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An operand of essentially signed or enum type should not be used as the right hand operand to the bitwise shifting operator" id="CODSTA-161\_f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An operand of essentially unsigned type should not be used as the operand to the unary minus operator" id="CODSTA-161\_g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations" id="CODSTA-162" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The value of an expression shall not be assigned to an object with a narrower essential type" id="CODSTA-163\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The value of an expression shall not be assigned to an object of a different essential type category" id="CODSTA-163\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category" id="CODSTA-164\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The second and third operands of the ternary operator shall have the same essential type category" id="CODSTA-164\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not cast from or to essentially Boolean type" id="CODSTA-165\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A loop counter in a 'for' loop shall not have essentially floating type" id="CODSTA-167\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A loop counter in 'while' and 'do-while' loops shall not have essentially floating type" id="CODSTA-167\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="There shall only be one loop counter in a 'for' loop, which shall not be modified in the 'for' loop body" id="CODSTA-168" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The first clause of a 'for' loop shall be well-formed" id="CODSTA-169\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The second clause of a 'for' loop shall be well-formed" id="CODSTA-169\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The third clause of a 'for' statement shall be well-formed" id="CODSTA-169\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use NULL identifier; use 0 instead" id="CODSTA-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The names of macros that exist prior to preprocessing should be distinct from the identifiers that exist after preprocessing (c90)" id="CODSTA-170\_c90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The names of macros that exist prior to preprocessing should be distinct from the identifiers that exist after preprocessing (c99)" id="CODSTA-170\_c99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The name of a macro should be distinct from the names of its parameters(c90)" id="CODSTA-171\_c90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The name of a macro should be distinct from the names of its parameters(c99)" id="CODSTA-171\_c99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The name of a macro should be distinct from the names of other macros that are currently defined(c90)" id="CODSTA-172\_c90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The name of a macro should be distinct from the names of other macros that are currently defined(c99)" id="CODSTA-172\_c99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A program should not exceed the translation limits imposed by The Standard (c90)" id="CODSTA-174\_a\_c90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A program should not exceed the translation limits imposed by The Standard (c99)" id="CODSTA-174\_a\_c99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A program should not exceed the translation limits imposed by The Standard (c90)" id="CODSTA-174\_b\_c90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A program should not exceed the translation limits imposed by The Standard (c99)" id="CODSTA-174\_b\_c99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="External identifiers shall be distinct" id="CODSTA-178" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Identifiers declared in the file scope and in the same name space shall be distinct (c90)" id="CODSTA-179\_a\_c90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Identifiers declared in the file scope and in the same name space shall be distinct (c99)" id="CODSTA-179\_a\_c99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Identifiers declared in the same block scope and name space shall be distinct (c90)" id="CODSTA-179\_b\_c90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Identifiers declared in the same block scope and name space shall be distinct (c99)" id="CODSTA-179\_b\_c99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Identifiers that define objects or functions with external linkage shall be unique" id="CODSTA-180" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The pointer arguments to the Standard Library functions 'memcmp', 'memmove' and 'memcmp' shall be pointers to qualified or unqualified versions of compatible types" id="CODSTA-183" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The pointer arguments to the Standard Library function 'memcmp' shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type" id="CODSTA-184" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Strings pointed by members of the structure 'lconv' should not be modified" id="CODSTA-185\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly" id="CODSTA-186" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Cast characters to unsigned char before assignment to larger integer sizes" id="CODSTA-187\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An expressions of the 'signed char' type should not be used as an array index" id="CODSTA-187\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Cast characters to unsigned char before converting to larger integer sizes" id="CODSTA-187\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not confuse narrow and wide character strings and functions" id="CODSTA-188" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not add or subtract a scaled integer to a pointer" id="CODSTA-189" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Use the ctype.h facilities for character test" id="CODSTA-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use object representations to compare floating-point values" id="CODSTA-190" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Wrap functions that can spuriously wake up in a loop" id="CODSTA-191" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The final member of a structure should not be an array of size '0' or '1'" id="CODSTA-192" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Allocate structures containing a flexible array member dynamically" id="CODSTA-193" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Wrap functions that can fail spuriously in a loop" id="CODSTA-194" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not refer to an atomic variable twice in an expression" id="CODSTA-195" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not access an array in the result of a function call" id="CODSTA-196" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The value of a complex expression of floating type should not be cast to a wider floating type" id="CODSTA-198" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The value of a complex expression of floating type should not be cast to an integer type" id="CODSTA-198\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use assertions" id="CODSTA-199" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Explicitly specify array bounds in array declarations with initializers" id="CODSTA-200" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not process structured text data natively" id="CODSTA-201" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Functions declared as 'noreturn' shall have the 'void' return type" id="CODSTA-204" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not cast an array to the pointer to a structure of a larger size than the size of the array" id="CODSTA-205" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The '\_Noreturn' function specifier should not be used" id="CODSTA-206" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The &lt;stdnoreturn.h> header file should not be used" id="CODSTA-207" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The &lt;stdalign.h> header file shall not be used" id="CODSTA-208" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The facilities that are specified as being provided by &lt;stdatomic.h> should not be used" id="CODSTA-209" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The '\_Thread\_local' storage class specifier should not be used" id="CODSTA-210" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The facilities that are specified as being provided by &lt;threads.h> should not be used" id="CODSTA-211" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The 'rsize\_t' type should not be used" id="CODSTA-212" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The '\_Alignas' alignment specifier and the '\_Alignof' operator should not be used" id="CODSTA-213" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The '\_Atomic' type specifier and the '\_Atomic' type qualifier should not be used" id="CODSTA-214" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The '\_\_STDC\_WANT\_LIB\_EXT1\_\_' macro should not be defined to the value other than '0'" id="CODSTA-215" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The '\_Generic' operator should not be used" id="CODSTA-216" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The 'errno\_t' type should not be used" id="CODSTA-217" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use following macros: RSIZE\_MAX, L\_tmpnam\_s, TMP\_MAX\_S" id="CODSTA-218" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use the functions defined in Annex K of ISO/IEC 9899:2011 standard" id="CODSTA-219" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Arguments of integer-constant macros should be decimal, octal, or hexadecimal constants with appropriate values" id="CODSTA-220" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not include any type qualifiers in the specification of a function type" id="CODSTA-221" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid implicit integral conversions from a wider to a narrower type" id="CODSTA-222" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Fixed width integer types from &lt;cstdint>, indicating the size and signedness, shall be used in place of the basic numerical types" id="CODSTA-223" sev="3">

<Stats authTot="20;" authUrg="0;" total="20"/>

</Rule>

<Rule cat="CODSTA" desc="The conditional operator should not be used as a sub-expression" id="CODSTA-224" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="All 'if' statements should have an 'else' clause" id="CODSTA-23" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid magic numbers" id="CODSTA-26" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid functions that modify global variables" id="CODSTA-27" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA" desc="&quot;#define&quot; or enum constants should be used instead of hard coded values whenever possible" id="CODSTA-29" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid returning handles to function parameters" id="CODSTA-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid explicit type conversions (casts)" id="CODSTA-31" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA" desc="Do not write logical expressions of the type if(test) or if(!test) when test is a pointer" id="CODSTA-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use operator ++ or -- in the conditional expression of if, while, or switch" id="CODSTA-33" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Use a typedef to simplify program syntax when declaring function pointers" id="CODSTA-34" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Always provide a default branch for switch statements" id="CODSTA-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Pass built-in-types by value unless you are modifying them" id="CODSTA-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Prefer const objects or enums to #defines for simple constants" id="CODSTA-37" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use a #define that prevents the compiler from checking types" id="CODSTA-38" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="'void' should be used when a function is passed or returns no values" id="CODSTA-40" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="All structures should be typedef'd" id="CODSTA-47" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use the following digraphs: &lt;%, %>, &lt;:, :>, %:, %:%:" id="CODSTA-48" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Null initialize or increment expressions in for loops will not be used; a while loop will be used instead" id="CODSTA-49" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Hexadecimal constants will be represented using all uppercase letters" id="CODSTA-50" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The initialization expression in a for loop will perform no actions other than to initialize the value of a single for loop parameter" id="CODSTA-52" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The increment expression in a for loop will perform no action other than to change a single loop parameter to the next value for the loop" id="CODSTA-53" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Every switch statement will have at least two cases and a potential default" id="CODSTA-54" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="All 'case' and 'default' labels of 'switch' statement should have an explicit 'break' or a 'return' statement, or 'fall through' comment" id="CODSTA-56" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid comparing values with TRUE macro/enum constant using equality operators (&quot;==&quot;, &quot;!=&quot;)" id="CODSTA-60" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The final clause of a switch statement shall be the default-clause unless all enumeration values are tested" id="CODSTA-61" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A cast shall not convert a pointer to a function to any other pointer type, including a pointer to function type" id="CODSTA-62" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Bitwise operators shall only be applied to operands of unsigned underlying type" id="CODSTA-63" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An unconditional throw or break statement shall terminate every non-empty switch-clause" id="CODSTA-64" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="An object with integer type or pointer to void type shall not be converted to an object with pointer type" id="CODSTA-65" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA" desc="Non-constant operands to a binary bitwise operator shall have the same underlying type" id="CODSTA-66" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The types used for an object, a function return type, or a function parameter shall be token-for-token identical in all declarations and re-declarations" id="CODSTA-67" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A &quot;U&quot; suffix shall be applied to all octal or hexadecimal integer literals of unsigned type" id="CODSTA-68" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Expressions with type (plain) char and wchar\_t shall not be used as operands to built-in operators other than =, ==, != and the unary &amp; operator" id="CODSTA-69" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Expressions with type enum shall not be used as operands to built-in operators other than [ ], =, ==, !=, &lt;, &lt;=, >, >=, and the unary &amp; operator" id="CODSTA-70" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Named bit-fields with signed integer type shall have a length of more than one bit" id="CODSTA-71" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Assembler instructions shall only be introduced using the asm declaration" id="CODSTA-73" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Bit-fields shall not have enum type" id="CODSTA-74" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Bit-fields shall be either bool type or an explicitly unsigned or signed integral type" id="CODSTA-75" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The identifier main shall not be used for a function other than the global function main" id="CODSTA-76" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The goto statement shall jump to a label declared later in the same function body" id="CODSTA-77" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Any label referenced by a goto statement shall be declared in the same block, or in a block enclosing the goto statement" id="CODSTA-78" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="For any iteration statement there shall be no more than one break or goto statement used for loop termination" id="CODSTA-79" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The continue statement shall only be used within a well formed for loop" id="CODSTA-80" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="If a function has internal linkage then all re-declarations shall include the 'static' storage class specifier" id="CODSTA-81" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid infinite loops" id="CODSTA-82" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use empty infinite loops" id="CODSTA-82\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="All loops must have a fixed upper or lower bound" id="CODSTA-83" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid exit points within infinite loops" id="CODSTA-85" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The validity of parameters must be checked inside each function" id="CODSTA-86" sev="3">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Rule>

<Rule cat="CODSTA" desc="Use no more than one level of dereferencing" id="CODSTA-87" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Function pointers are not permitted" id="CODSTA-88" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The declaration should not contain more than one level of pointer indirection" id="CODSTA-89" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CODSTA" desc="Each operand of a logical '&amp;&amp;' or '||' shall be a postfix-expression" id="CODSTA-90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A function shall have at most one exit point" id="CODSTA-91" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The names of standard library macros and objects shall not be reused" id="CODSTA-92" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined (C90 code)" id="CODSTA-92\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined (C99 code)" id="CODSTA-92\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not declare pointer or array type" id="CODSTA-94" sev="3">

<Stats authTot="30;" authUrg="0;" total="30"/>

</Rule>

<Rule cat="CODSTA" desc="Do not declare pointer type" id="CODSTA-95" sev="3">

<Stats authTot="30;" authUrg="0;" total="30"/>

</Rule>

<Rule cat="CODSTA" desc="Bit-fields shall have explicitly unsigned integral or enumeration types only" id="CODSTA-96" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Use parenthesis to clarify expression order if operators with precedence lower than arithmetic are used" id="CODSTA-97" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="All 'if...else if' constructs will contain either a final else clause or a comment indicating why a final else clause is not necessary" id="CODSTA-98" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Signed and unsigned values shall not be mixed in comparison operations" id="CODSTA-99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The second or third operand of a ternary operator '?:' shall not contain side effects" id="CODSTA-105" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The exception handling features of &lt;fenv.h> should not be used" id="CODSTA-109" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden" id="CODSTA-115" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects" id="CODSTA-123" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A conversion should not be performed between a pointer to object and an integer type" id="CODSTA-127" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A conversion should not be performed between a pointer to object type and an integer type other than 'uintptr\_t' or 'intptr\_t'" id="CODSTA-127\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A conversion should not be performed from pointer to void into pointer to object" id="CODSTA-128" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A function parameter should not be modified" id="CODSTA-132" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements" id="CODSTA-134" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use enumerations in boolean condition" id="CODSTA-151" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The cast operation to essentially enumeration type is not allowed" id="CODSTA-165\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not use casts between essentially character types and essentially floating types" id="CODSTA-165\_c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A function should not contain unused type declarations" id="CODSTA-175\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A source file should not contain unused type declarations" id="CODSTA-175\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A function should not contain unused local tag declarations" id="CODSTA-176\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A source file should not contain unused tag declarations" id="CODSTA-176\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="A source file should not contain unused macro definitions" id="CODSTA-177" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The +, -, += and -= operators should not be applied to an expression of pointer type" id="CODSTA-181" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Avoid switch statements with only one case" id="CODSTA-41" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Suspicious use of semicolon" id="CODSTA-57" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Cast to void is not allowed" id="CODSTA-58" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Hardcoded array declarations and 'malloc' calls should not be used" id="CODSTA-59" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not declare member variables as bit-fields" id="CODSTA-02" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not declare local variables with the 'static' keyword" id="CODSTA-04" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Assert liberally to document internal assumptions and invariants" id="CODSTA-11" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Assertions should not contain function calls nor function-like macro calls" id="CODSTA-150\_b" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Prefer while statements over do statements" id="CODSTA-18" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Do not hard code string literals" id="CODSTA-203" sev="5">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CODSTA" desc="When using enum, the values of each member should be explicitly declared" id="CODSTA-21" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The macro FALSE should be defined as 0 (zero)" id="CODSTA-24" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The enumeration constant named 'FALSE' should be explicitly initialized by 0 value" id="CODSTA-25" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Define fields for union declarations" id="CODSTA-28" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The macro TRUE should be defined as 1" id="CODSTA-42" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="The enumeration constant named 'TRUE' should be explicitly initialized by 1 value" id="CODSTA-43" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA" desc="Use positive logic rather than negative logic whenever practical" id="CODSTA-46" sev="5">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CODSTA" desc="The names of standard library functions shall not be overridden" id="CODSTA-93" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Constructors allowing for conversion should be made explicit" id="CODSTA-CPP-04" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not use user-defined conversion functions" id="CODSTA-CPP-05" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The first operand of a conditional-operator shall have type bool" id="CODSTA-CPP-65" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Declare at least one constructor to prevent the compiler from doing so" id="CODSTA-CPP-19" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Arrays shall not be used in interfaces" id="CODSTA-CPP-55" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Private and protected methods shall not declare parameters with array type" id="CODSTA-CPP-55\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Copy operations must not mutate the source object" id="CODSTA-CPP-98" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Have assignment operator returns a reference to \*this; make assignment operator's return type a non-const reference to it's class' type" id="CODSTA-CPP-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Bitwise operators, comparison operators, logical operators, comma operator should be const" id="CODSTA-CPP-03" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Avoid returning handles to class data from member functions" id="CODSTA-CPP-06" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Postfix increment and decrement should be implemented in terms of their prefix counterparts" id="CODSTA-CPP-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Avoid overloading logical operators AND, OR (&amp;&amp;, ||)" id="CODSTA-CPP-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Avoid using reinterpret\_cast" id="CODSTA-CPP-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not define structs that contain member functions" id="CODSTA-CPP-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not compare objects of a class that may contain padding bits with C standard library functions" id="CODSTA-CPP-100" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="A relational operator shall return a boolean value" id="CODSTA-CPP-101" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="A binary arithmetic operator and a bitwise operator shall return a 'prvalue'" id="CODSTA-CPP-102" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Output parameters shall not be used" id="CODSTA-CPP-103" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The operand of the 'typeid' operator shall not contain any expression that has side effects" id="CODSTA-CPP-104" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The operand of the 'typeid' operator shall not contain a function call that causes side effects" id="CODSTA-CPP-104\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Define special members as =default when the behavior is equivalent to the compiler's behavior" id="CODSTA-CPP-105" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Use delegating constructors to reduce code duplication" id="CODSTA-CPP-106" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Comparison operators shall be non-member functions" id="CODSTA-CPP-107" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Comparison operators shall be non-member functions with identical parameter types and noexcept" id="CODSTA-CPP-107\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Prefer C++-style casts" id="CODSTA-CPP-11" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Put classes that are used as base classes and classes that are used as member variables into separate include files" id="CODSTA-CPP-12" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Put classes that are used as function return type into separate include files" id="CODSTA-CPP-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Put classes that are used as argument types in function prototypes into separate include files" id="CODSTA-CPP-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Put functions that are used in the body of inline member function into separate include file" id="CODSTA-CPP-15" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="When two operators are opposites (such as == and !=), it is appropriate to define both" id="CODSTA-CPP-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not use the 'struct' keyword to declare a variable in C++" id="CODSTA-CPP-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="If you'd like to support mixed-mode operations make operators a non-member functions" id="CODSTA-CPP-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Avoid making any assignment operator virtual. Do not return const T&amp; from assignment operator" id="CODSTA-CPP-24" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Consider making virtual functions nonpublic, and public functions nonvirtual" id="CODSTA-CPP-25" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Keep types and functions in separate namespaces unless they're specifically intended to work together" id="CODSTA-CPP-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Keep a type and its nonmember function interface in the same namespace" id="CODSTA-CPP-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="When binary arithmetic operators are defined ( + , - , \* , / , ^ , % , | , &amp; ), assignment versions should be provided too" id="CODSTA-CPP-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Prefer the canonical forms of arithmetic and assignment operators" id="CODSTA-CPP-29" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Prefer non-member operators than member ones to support mixed-mode arithmetic" id="CODSTA-CPP-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="A function definition should not be placed in a class specification unless the function is intended to be inlined" id="CODSTA-CPP-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Member functions shall not be defined within the template class definition" id="CODSTA-CPP-33" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Avoid using static\_cast on pointers" id="CODSTA-CPP-34" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Avoid dynamic\_casts" id="CODSTA-CPP-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The global namespace shall only contain main() and namespace declarations" id="CODSTA-CPP-36" sev="3">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not define class/struct/union inside function implementation" id="CODSTA-CPP-37" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Conversion operator, operator->, operator(), operator[] should be const" id="CODSTA-CPP-38" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Don't write namespace usings in a header file or before an #include" id="CODSTA-CPP-39" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not use the 'enum' keyword to declare a variable in C++" id="CODSTA-CPP-41" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not declare member variables with the 'mutable' keyword" id="CODSTA-CPP-42" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Declare reference parameters as const references whenever possible" id="CODSTA-CPP-43" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Have the non-const version call the const version of member function instead of duplicating the const version definition" id="CODSTA-CPP-44" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="In the private section of a class items shall be declared in the following order: Constructors, Destructor, Member Functions, Member Operator Function, Enumerations and others" id="CODSTA-CPP-45" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Order of scopes in class: public before all others" id="CODSTA-CPP-46" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Order of scopes in classes: protected before private" id="CODSTA-CPP-47" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="In the protected section of a class items shall be declared in the following order: Constructors, Destructor, Member Functions, Member Operator Function, Enumerations and others" id="CODSTA-CPP-48" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="In the public section of a class items shall be declared in the following order: Constructors, Destructor, Member Functions, Member Operator Function, Enumerations and others" id="CODSTA-CPP-49" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not use static keyword except inside functions and classes" id="CODSTA-CPP-50" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not define inline functions in source files" id="CODSTA-CPP-51" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Consider using the natural relationship between the assignment version of an operator and the stand-alone version" id="CODSTA-CPP-52" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Declare parameters or local variable as const whenever possible" id="CODSTA-CPP-53" sev="3">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Member functions shall be declared const whenever possible" id="CODSTA-CPP-54" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="A class, structure, or enumeration will not be declared in the definition of its type" id="CODSTA-CPP-56" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Namespaces will not be nested more than two levels deep" id="CODSTA-CPP-57" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The value returned by a function having a non-void return type that is not an overloaded operator shall always be used" id="CODSTA-CPP-58" sev="3">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The C library shall not be used" id="CODSTA-CPP-59" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Only those escape sequences that are defined in ISO/IEC 14882:2003 shall be used" id="CODSTA-CPP-60" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Only those escape sequences that are defined in ISO/IEC 14882:2014 shall be used" id="CODSTA-CPP-60\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="NULL shall not be used as an integer value" id="CODSTA-CPP-62" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Literal zero (0) shall not be used as the null-pointer-constant" id="CODSTA-CPP-63" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The condition of an if-statement and the condition of an iteration-statement shall have type bool" id="CODSTA-CPP-64" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="C-style casts (other than void casts) and functional notation casts (other than explicit constructor calls) shall not be used" id="CODSTA-CPP-66" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Each operand of the ! operator, the logical &amp;&amp; or the logical || operators shall have type bool" id="CODSTA-CPP-67" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The unary &amp; operator shall not be overloaded" id="CODSTA-CPP-68" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="A for loop shall contain a single loop-counter which shall not have floating type" id="CODSTA-CPP-69" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="If loop-counter is not modified by -- or ++, then, within condition, the loop-counter shall only be used as an operand to &lt;=, &lt;, > or >=" id="CODSTA-CPP-70" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The loop-counter shall be modified by one of: --, ++, -=n, or +=n; where n remains constant for the duration of the loop" id="CODSTA-CPP-71" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="A loop-control-variable other than the loop-counter shall not be modified within condition or expression" id="CODSTA-CPP-72" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="A loop-control-variable other than the loop-counter which is modified in statement shall have type bool" id="CODSTA-CPP-73" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="There shall be no unnamed namespaces in header files" id="CODSTA-CPP-74" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="using-directives shall not be used" id="CODSTA-CPP-75" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Multiple declarations for an identifier in the same namespace shall not straddle a using-declaration for that identifier" id="CODSTA-CPP-76" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Const member functions shall not return non-const pointers or references to class-data" id="CODSTA-CPP-77" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="If a member function can be made static then it shall be made static, otherwise if it can be made const then it shall be made const" id="CODSTA-CPP-78" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Neither operand of an equality operator (== or !=) shall be a pointer to a virtual member function" id="CODSTA-CPP-79" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Avoid overloading comma operator &quot;,&quot;" id="CODSTA-CPP-80" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Avoid unencapsulated global variables (including variables declared in namespaces and public static members)" id="CODSTA-CPP-82" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Identifiers declared in a local scope should not hide identifiers declared in a class scope" id="CODSTA-CPP-83" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Identifiers declared in a class scope should not hide identifiers declared in a global or namespace scope" id="CODSTA-CPP-84" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Identifiers declared in an inner class scope should not hide identifiers declared in outer class scope" id="CODSTA-CPP-85" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="If two opposite equality operators ('==', '!=') are defined in a class, one shall be defined in terms of the other" id="CODSTA-CPP-86" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="CODSTA-CPP-87\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="CODSTA-CPP-87\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="CODSTA-CPP-87\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The name of typedef should not be the same as the name of its basic type" id="CODSTA-CPP-88" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="using-directives and using-declarations (excluding class scope or function scope using-declarations) shall not be used in header files" id="CODSTA-CPP-90" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="The overloaded binary operator should be implemented in terms of its corresponding compound assignment operator" id="CODSTA-CPP-91" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="All accessible entity names within a multiple inheritance hierarchy should be unique" id="CODSTA-CPP-92" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not initialize objects with a non-trivial class type using C standard library functions" id="CODSTA-CPP-93" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not compare objects of nonstandard-layout class type with C standard library functions" id="CODSTA-CPP-94" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not modify the standard namespaces 'std' and 'posix'" id="CODSTA-CPP-95" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not call a function with a mismatched language linkage" id="CODSTA-CPP-96" sev="3">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Never qualify a reference type with 'const' or 'volatile'" id="CODSTA-CPP-97" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Use offsetof() on valid types and members" id="CODSTA-CPP-99" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Prefer non-member non-friend functions to member functions" id="CODSTA-CPP-22" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Classes which have only getters/setters (accessors/mutators) are not allowed" id="CODSTA-CPP-81" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not assign enumerations to bool" id="CODSTA-CPP-89" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Prefer iostream.h to stdio.h" id="CODSTA-CPP-01" sev="5">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Encapsulate global variables and constants, enumerated types, and typedefs in a class" id="CODSTA-CPP-18" sev="5">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Assignment operator must return const reference" id="CODSTA-CPP-21" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Whenever a global function is referenced, use the :: operator" id="CODSTA-CPP-23" sev="5">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Do not use the keyword 'explicit' for a constructor" id="CODSTA-CPP-31" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-CPP" desc="Limiting the number of objects of a class" id="CODSTA-CPP-40" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer lambdas over std::bind, std::bind1st and std::bind2nd" id="CODSTA-MCPP-07" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer 'auto' to explicit type names in variable declarations" id="CODSTA-MCPP-08\_a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer 'auto' to explicit type names for function return types" id="CODSTA-MCPP-08\_b" sev="2">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="All std::hash specializations for user-defined types shall have a noexcept function call operator" id="CODSTA-MCPP-27" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="User-conversion cast operators should be made explicit" id="CODSTA-MCPP-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer alias declarations to typedefs" id="CODSTA-MCPP-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer Scoped Enums to Unscoped Enums" id="CODSTA-MCPP-03" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Each overriding virtual function shall be declared with the override or final specifier" id="CODSTA-MCPP-05" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Declare copy constructor and copy assignment operators with the 'delete' specifier to prevent copying of class" id="CODSTA-MCPP-06\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Declare copy constructor and copy assignment operators with the 'delete' specifier instead of using a base class with private methods to prevent copying of class" id="CODSTA-MCPP-06\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="The 'binder1st' and 'binder2nd' identifiers should not be used" id="CODSTA-MCPP-07\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Declare functions 'noexcept' if they will not emit exceptions" id="CODSTA-MCPP-09" sev="3">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer const iterators to iterators" id="CODSTA-MCPP-10\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer to use cbegin(), crbegin, cend(), crend() functions" id="CODSTA-MCPP-10\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Use std::move() on rvalue references and std::forward() on forwarding references" id="CODSTA-MCPP-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="The 'std::forward' function shall be used to forward universal references" id="CODSTA-MCPP-13\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Avoid Overloading on Forwarding References" id="CODSTA-MCPP-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Avoid default capture modes" id="CODSTA-MCPP-15\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Use the 'this' pointer explicitly in lambdas with a default by-reference capture" id="CODSTA-MCPP-15\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer 'std::make\_shared' to the direct use of new" id="CODSTA-MCPP-16\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer to use std::unique\_ptr instead of std::auto\_ptr" id="CODSTA-MCPP-16\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="'std::make\_unique' shall be used to construct objects owned by 'std::unique\_ptr'" id="CODSTA-MCPP-16\_e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Never return lambdas that capture local objects by reference" id="CODSTA-MCPP-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Never capture local objects from an outer lambda by reference" id="CODSTA-MCPP-17\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="The lambda that captures local objects by reference should not be assigned to the variable with a greater lifetime" id="CODSTA-MCPP-17\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Avoid unnecessary default capture modes in lambda expressions" id="CODSTA-MCPP-18\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Avoid unnecessary lambda captures" id="CODSTA-MCPP-18\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Declare assignment operators with the ref-qualifier &amp;" id="CODSTA-MCPP-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Use explicit ref-qualifiers on auto declarations in range-based 'for' loops" id="CODSTA-MCPP-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="If a public destructor of a class is non-virtual, then the class should be declared final" id="CODSTA-MCPP-23" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Only one of virtual, override or final should be specified in a member function declaration" id="CODSTA-MCPP-24" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Digit sequences separators ' shall only be used consistently" id="CODSTA-MCPP-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="A pointer to member virtual function shall only be tested for equality with null-pointer-constant" id="CODSTA-MCPP-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="A lambda shall not be an operand to typeid" id="CODSTA-MCPP-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Use smart pointers when passing a pointer to an object in a thread" id="CODSTA-MCPP-29" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Consider using 'std::unique\_ptr' instead of 'std::shared\_ptr' for local objects" id="CODSTA-MCPP-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Define both sized and unsized versions of operator delete" id="CODSTA-MCPP-31" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Static and thread-local objects shall be constant-initialized" id="CODSTA-MCPP-32" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="User defined literals operators shall only perform conversion of passed parameters" id="CODSTA-MCPP-33" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="A smart pointer shall only be used as a parameter type if it expresses lifetime semantics" id="CODSTA-MCPP-34" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="A parameter should only be declared as a non-const lvalue reference to 'std::shared\_ptr' or 'std::unique\_ptr' if the function replaces the managed object" id="CODSTA-MCPP-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Do not declare the type of a parameter as an rvalue reference to 'std::shared\_ptr' or 'std::unique\_ptr'" id="CODSTA-MCPP-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Derived classes that do not need further explicit initialization and require all the constructors from the base class shall use inheriting constructors" id="CODSTA-MCPP-37" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Braced-initialization {}, without equals sign, shall be used for variable initialization" id="CODSTA-MCPP-38" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="A variable of type auto shall not be initialized using '{}' or '={}' braced-initialization" id="CODSTA-MCPP-39" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Do not overuse 'auto' specifier" id="CODSTA-MCPP-40" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="The 'random\_shuffle' identifier should not be used" id="CODSTA-MCPP-41" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Do not use the increment operator (++) on an operand of type 'bool'" id="CODSTA-MCPP-42" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="The 'set\_unexpected' identifier should not be used" id="CODSTA-MCPP-43" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Lambda expressions should not be defined inside another lambda expression" id="CODSTA-MCPP-44" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Return type of a non-void return type lambda expression should be explicitly specified" id="CODSTA-MCPP-45" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Include a parameter list in every lambda expression" id="CODSTA-MCPP-46" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Avoid overloading constructors with std::initializer\_list" id="CODSTA-MCPP-47" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Multiple output values from a function should be returned as a struct or tuple" id="CODSTA-MCPP-48" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="User-defined copy and move assignment operators should use user-defined no-throw swap function" id="CODSTA-MCPP-49" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="A for-loop that loops through all elements of the container and does not use its loop-counter shall not be used" id="CODSTA-MCPP-50" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Use const container calls when the result is immediately converted to a const iterator" id="CODSTA-MCPP-51" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Do not introduce virtual functions in a final class" id="CODSTA-MCPP-52" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Do not use std::move on objects declared with the const or const &amp; type" id="CODSTA-MCPP-53" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Do not create an rvalue reference of std::array" id="CODSTA-MCPP-54" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Use std::call\_once rather than the Double-Checked Locking pattern" id="CODSTA-MCPP-55" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer 'nullptr' over 'NULL' or '0'(zero)" id="CODSTA-MCPP-04" sev="4">

<Stats authTot="21;" authUrg="0;" total="21"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Use constexpr to declare const variables whenever possible" id="CODSTA-MCPP-11\_a\_cpp11" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Use constexpr to declare functions whenever possible" id="CODSTA-MCPP-11\_b\_cpp11" sev="4">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer smart pointer members over raw pointer members" id="CODSTA-MCPP-16\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer smart pointers over raw pointers for arrays or STL containers" id="CODSTA-MCPP-16\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Prefer smart pointers over raw local pointers" id="CODSTA-MCPP-20" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CODSTA-MCPP" desc="Do not call lock() directly on a mutex" id="CODSTA-MCPP-21" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="COMMENT" desc="Line-splicing shall not be used in // comments" id="COMMENT-13" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="COMMENT" desc="Prefer C++ style comment" id="COMMENT-01" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="COMMENT" desc="Provide copyright information" id="COMMENT-02" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="COMMENT" desc="Every source file will be documented with an introductory comment that provides information on the file" id="COMMENT-03" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="COMMENT" desc="Document functions in comments that precede function definitions" id="COMMENT-04" sev="3">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Rule>

<Rule cat="COMMENT" desc="Document functions in comments that precede function declarations" id="COMMENT-04\_b" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="COMMENT" desc="Each variable declaration should be commented" id="COMMENT-05" sev="3">

<Stats authTot="21;" authUrg="0;" total="21"/>

</Rule>

<Rule cat="COMMENT" desc="Each typedef should be commented" id="COMMENT-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="COMMENT" desc="Each enumeration value should be commented" id="COMMENT-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="COMMENT" desc="Each structure member variable should be commented" id="COMMENT-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="COMMENT" desc="All usage of assembler shall be documented" id="COMMENT-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="COMMENT" desc="Use of floating-point arithmetic shall be documented" id="COMMENT-10" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="COMMENT" desc="The character sequence // shall not be used within a C-style comment" id="COMMENT-11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="COMMENT" desc="The character sequence /\* shall not be used within a C++-style comment" id="COMMENT-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="COMMENT" desc="All declarations of types, data members, and functions should be preceded by a comment annotated with the '@brief' tag" id="COMMENT-14" sev="3">

<Stats authTot="33;" authUrg="0;" total="33"/>

</Rule>

<Rule cat="COMMENT" desc="Function parameters and return type should be documented in a comment that precedes the function declaration" id="COMMENT-14\_b" sev="3">

<Stats authTot="32;" authUrg="0;" total="32"/>

</Rule>

<Rule cat="CWE-119" desc="Avoid buffer overflow due to defining incorrect format limits" id="CWE-119-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Avoid overflow when reading from a buffer" id="CWE-119-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Avoid overflow when writing to a buffer" id="CWE-119-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Avoid tainted data in array indexes" id="CWE-119-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Prevent buffer overflows from tainted data" id="CWE-119-g" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Avoid buffer read overflow from tainted data" id="CWE-119-h" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Avoid buffer write overflow from tainted data" id="CWE-119-i" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Avoid accessing arrays out of bounds" id="CWE-119-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Avoid accessing arrays and pointers out of bounds" id="CWE-119-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Suspicious use of 'strcpy' without checking size of source buffer" id="CWE-119-j" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-119" desc="Ensure the output buffer is large enough when using path manipulation functions" id="CWE-119-k" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-125" desc="Avoid overflow when reading from a buffer" id="CWE-125-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-125" desc="Avoid buffer read overflow from tainted data" id="CWE-125-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-125" desc="Avoid accessing arrays out of bounds" id="CWE-125-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-125" desc="Avoid accessing arrays and pointers out of bounds" id="CWE-125-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-190" desc="Avoid integer overflows" id="CWE-190-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-190" desc="Avoid possible integer overflow in expressions in which the result is cast to a wider integer type" id="CWE-190-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-190" desc="Avoid possible integer overflow in expressions in which the result is assigned to a variable of a wider integer type" id="CWE-190-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-190" desc="Avoid possible integer overflow in expressions in which the result is compared to an expression of a wider integer type" id="CWE-190-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-190" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="CWE-190-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-190" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="CWE-190-f" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-190" desc="Evaluation of constant unsigned integer expressions should not lead to wrap-around" id="CWE-190-g" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Avoid tainted data in array indexes" id="CWE-20-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Protect against integer overflow/underflow from tainted data" id="CWE-20-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Protect against command injection" id="CWE-20-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Avoid printing tainted data on the output console" id="CWE-20-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Protect against environment injection" id="CWE-20-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Exclude unsanitized user input from format strings" id="CWE-20-g" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Protect against SQL injection" id="CWE-20-h" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Protect against file name injection" id="CWE-20-i" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Avoid passing unvalidated binary data to log methods" id="CWE-20-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-20" desc="Untrusted data is used as a loop boundary" id="CWE-20-j" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CWE-200" desc="Do not print potentially sensitive information, resulting from an application error into exception messages" id="CWE-200-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-22" desc="Protect against file name injection" id="CWE-22-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-269" desc="Observe correct revocation order while relinquishing privileges" id="CWE-269-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-269" desc="Ensure that privilege relinquishment is successful" id="CWE-269-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-287" desc="Do not use weak encryption functions" id="CWE-287-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-326" desc="Do not use weak encryption functions" id="CWE-326-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-362" desc="Avoid race conditions when using fork and file descriptors" id="CWE-362-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-362" desc="Do not use global variable with different locks set" id="CWE-362-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-362" desc="Usage of functions prone to race is not allowed" id="CWE-362-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-362" desc="Avoid race conditions while accessing files" id="CWE-362-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-362" desc="Use locks to prevent race conditions when modifying bit fields" id="CWE-362-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-400" desc="Do not create variables on the stack above the defined limits" id="CWE-400-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-415" desc="Do not use resources that have been freed" id="CWE-415-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-416" desc="Do not use resources that have been freed" id="CWE-416-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-416" desc="Do not point to a wrapped object that has been freed" id="CWE-416-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-416" desc="Freed memory shouldn't be accessed under any circumstances" id="CWE-416-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-426" desc="Use care to ensure that LoadLibrary() will load the correct library" id="CWE-426-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-476" desc="Avoid null pointer dereferencing" id="CWE-476-a" sev="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CWE-476" desc="Do not check for null after dereferencing" id="CWE-476-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-532" desc="Avoid passing sensitive data to functions that write to log files" id="CWE-532-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-611" desc="Disable resolving XML external entities (XXE) in libxerces-c" id="CWE-611-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-617" desc="Do not use assertions" id="CWE-617-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="Conversions shall not be performed between a pointer to a function and any other type than pointer to function" id="CWE-704-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="Conversions shall not be performed between non compatible pointer to a function types" id="CWE-704-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="Conversions shall not be performed between a pointer to an incomplete type and any other type" id="CWE-704-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="A cast shall not be performed between a pointer to object type and a pointer to a different object type" id="CWE-704-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="A cast shall not be performed between pointer to void and an arithmetic type" id="CWE-704-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="An implicit conversion shall not be performed between pointer to void and an arithmetic type" id="CWE-704-h" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="A cast shall not be performed between pointer to object and a non-integer arithmetic type" id="CWE-704-i" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="Implicit conversions from wider to narrower integral type which may result in a loss of information shall not be used" id="CWE-704-j" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="Implicit conversions from integral to floating type which may result in a loss of information shall not be used" id="CWE-704-k" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="Implicit conversions from integral constant to floating type which may result in a loss of information shall not be used" id="CWE-704-l" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="A conversion should not be performed between a pointer to object type and an integer type other than 'uintptr\_t' or 'intptr\_t'" id="CWE-704-e" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-704" desc="A conversion should not be performed from pointer to void into pointer to object" id="CWE-704-f" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-732" desc="Specify the access permission bits if a file is created using the 'open' or 'openat' system call" id="CWE-732-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-732" desc="Call 'umask' before calling 'mkstemp'" id="CWE-732-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-770" desc="Validate potentially tainted data before it is used to determine the size of memory allocation" id="CWE-770-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CWE-772" desc="Ensure resources are freed" id="CWE-772-a" sev="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="CWE-772" desc="Define a virtual destructor in classes used as base classes which have virtual functions" id="CWE-772-b" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="CWE-78" desc="Protect against command injection" id="CWE-78-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-787" desc="Avoid buffer overflow due to defining incorrect format limits" id="CWE-787-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-787" desc="Avoid overflow when writing to a buffer" id="CWE-787-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-787" desc="Prevent buffer overflows from tainted data" id="CWE-787-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-787" desc="Avoid buffer write overflow from tainted data" id="CWE-787-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-787" desc="Avoid accessing arrays out of bounds" id="CWE-787-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-787" desc="Avoid accessing arrays and pointers out of bounds" id="CWE-787-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-787" desc="Ensure the output buffer is large enough when using path manipulation functions" id="CWE-787-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-798" desc="Do not hard code string literals" id="CWE-798-a" sev="5">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="CWE-835" desc="Avoid infinite loops" id="CWE-835-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-863" desc="Do not use 'cuserid' function" id="CWE-863-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="CWE-89" desc="Protect against SQL injection" id="CWE-89-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Never allow an exception to be thrown from a destructor, deallocation, and swap" id="EXCEPT-01" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Throw by value, catch by reference" id="EXCEPT-02" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Do not throw from within destructor" id="EXCEPT-03" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="The assignment-expression of a throw statement shall not itself cause an exception to be thrown" id="EXCEPT-11" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="All exceptions should be rethrown or logged with standard logger" id="EXCEPT-04" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="C++ exceptions shall not be used (i.e. throw, catch and try shall not be used.)" id="EXCEPT-05" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="EXCEPT" desc="There should be at least one exception handler to catch all otherwise unhandled exceptions" id="EXCEPT-06" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="EXCEPT" desc="An empty throw (throw;) shall only be used in the compound-statement of a catch handler" id="EXCEPT-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Exceptions shall be raised only after start-up and before termination of the program" id="EXCEPT-08" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="EXCEPT" desc="An exception object should not have pointer type" id="EXCEPT-09" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="EXCEPT" desc="Control shall not be transferred into a try or catch block using a goto or a switch statement" id="EXCEPT-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="NULL shall not be thrown explicitly" id="EXCEPT-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="EXCEPT-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Where a function's declaration includes an exception-specification, the function shall only be capable of throwing exceptions of the indicated type(s)" id="EXCEPT-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="A class type exception shall always be caught by reference" id="EXCEPT-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Handlers of a function-try-block implementation of a class constructor or destructor shall not reference nonstatic members from this class or its bases" id="EXCEPT-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class" id="EXCEPT-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Function called in global or namespace scope shall not throw unhandled exceptions" id="EXCEPT-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Exception objects must be nothrow copy constructible" id="EXCEPT-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="An explicitly declared copy constructor for a class that inherits from 'std::exception' should have a non-throwing exception specification" id="EXCEPT-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="All user-provided move constructors and move assignment operators shall not exit with an exception" id="EXCEPT-21" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Checked exceptions that could be thrown from a function shall be specified in the comment directly before the function declaration" id="EXCEPT-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Do not use throw exception specifications" id="EXCEPT-23" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Where multiple handlers are provided in a single 'try-catch' statement or 'function-try-block', any ellipsis (catch-all) handler shall occur last" id="EXCEPT-24" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Do not leave 'catch' blocks empty" id="EXCEPT-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="EXCEPT" desc="Avoid using catch-all exception handlers" id="EXCEPT-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Place an opening brace '{' on its own line" id="FORMAT-02" sev="3">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Rule>

<Rule cat="FORMAT" desc="Place a closing brace '}' on its own line" id="FORMAT-03" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="FORMAT" desc="Physical lines should be less than eighty characters" id="FORMAT-04" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="FORMAT" desc="The length of a macro should not exceed 10 lines" id="FORMAT-05" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Only one statement shall be allowed per line" id="FORMAT-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character preceding assignment operators" id="FORMAT-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character following assignment operators" id="FORMAT-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character preceding bitwise operators" id="FORMAT-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character following bitwise operators" id="FORMAT-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character preceding and following bitwise operator '&amp;'" id="FORMAT-11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character between a conditional statement keyword and its opening parenthesis" id="FORMAT-12" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a maximum of 1 ASCII space character following the opening parenthesis in conditional statements" id="FORMAT-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character preceding ternary conditional operator" id="FORMAT-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character following ternary conditional operator" id="FORMAT-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character preceding and following relational and equality operators" id="FORMAT-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be no white space following '.' or '->' operator" id="FORMAT-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be no white space preceding '.' or '->' operator" id="FORMAT-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character following all commas" id="FORMAT-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character following all semicolons" id="FORMAT-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There should be no space between a unary operator &quot;!&quot; or &quot;~&quot; and its operand" id="FORMAT-21" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There should be no space between a increment/decrement operator (++/--) and its operand" id="FORMAT-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be no white spaces between the 'return' statement or the 'sizeof' operator and the following opening parenthesis" id="FORMAT-23" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be no spaces after the opening '(' and before closing ')' parenthesis that enclose the operand of the 'return' statement or the 'sizeof' operator" id="FORMAT-24" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="The operand of the 'sizeof' operator should be enclosed in parentheses" id="FORMAT-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Parenthesis shall be used with the &quot;return&quot; statement" id="FORMAT-25\_b" sev="3">

<Stats authTot="12;" authUrg="0;" total="12"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be a single ASCII space character preceding and following logical operators" id="FORMAT-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Line should be indented by a multiple of four spaces" id="FORMAT-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="In a function definition, the return type of the function should be written on a separate line directly above the function name" id="FORMAT-28" sev="3">

<Stats authTot="14;" authUrg="0;" total="14"/>

</Rule>

<Rule cat="FORMAT" desc="Multiple variable declarations shall not be allowed on the same line" id="FORMAT-29" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="FORMAT" desc="Place left parenthesis directly after function name" id="FORMAT-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Separate logical tests in conditional expressions" id="FORMAT-31" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Each variable should be declared in a separate declaration statement" id="FORMAT-33" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="FORMAT" desc="Braces &quot;{}&quot; which enclose a block should be placed in the same column" id="FORMAT-34" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="FORMAT" desc="When declaring functions, the leading parenthesis and the first argument are to be written on the same line as the function name" id="FORMAT-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Sibling statement lines should be indented to the same level" id="FORMAT-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="First line in control statement body should be indented more than control statement keyword" id="FORMAT-37" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="When declaring functions with more than 2 parameters, the leading parenthesis and the first argument are to be written on the same line as the function name, each additional argument will be written on a separate line" id="FORMAT-38" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Braces (&quot;{}&quot;) which enclose a block will have nothing else on the line except comments" id="FORMAT-42" sev="3">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Rule>

<Rule cat="FORMAT" desc="Braces (&quot;{}&quot;) which enclose a block will be placed in the same column" id="FORMAT-43" sev="3">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Rule>

<Rule cat="FORMAT" desc="Braces (&quot;{}&quot;) which enclose a block will not have an empty line after &quot;{&quot; nor before &quot;}&quot;" id="FORMAT-44" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There should be no space between a unary operator &quot;&amp;&quot;, &quot;\*&quot;, &quot;+&quot;, &quot;-&quot; and its operand" id="FORMAT-45" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There should be no space between an increment/decrement operator (++/--) and its operand in macro definition" id="FORMAT-46" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Place CV-qualifiers on the right hand side of the type they apply to" id="FORMAT-47\_a" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="FORMAT" desc="CV-qualifiers shall be placed on the right hand side of the type that is a typedef or a using name" id="FORMAT-47\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Parameter names in function declarations should not be enclosed in parentheses" id="FORMAT-48" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Local variable names in variable declarations should not be enclosed in parentheses" id="FORMAT-49" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="The dereference operator '\*' and the address-of operator '&amp;' should be directly connected with the type" id="FORMAT-32" sev="4">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="FORMAT" desc="Sort #include directives in alphabetical order" id="FORMAT-39" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="Tabs that do not use ASCII spaces shall not be used" id="FORMAT-01" sev="5">

<Stats authTot="160;" authUrg="0;" total="160"/>

</Rule>

<Rule cat="FORMAT" desc="White spaces after the opening square bracket '[' and before its closing square bracket ']' shall be used in consistent way" id="FORMAT-40" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="FORMAT" desc="There shall be no space between '[' opening square bracket and preceding token" id="FORMAT-41" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="All declarations of an object or function shall have compatible types" id="GLOBAL-COMPATDECLS" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="Do not use more than one mutex for concurrent waiting operations on a condition variable" id="GLOBAL-CONDMUTEXVAR" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="If a function is declared with an exception-specification, then all declarations of the same function (in other translation units) shall be declared with the same set of type-ids" id="GLOBAL-EXCSPECDECL" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="An inline function that is used in multiple translation units shall be defined in one and only one file" id="GLOBAL-ONEDEFINLINE" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="The One Definition Rule shall not be violated" id="GLOBAL-ONEDEFRULE" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="A function template that is used in multiple translation units shall be defined in one and only one file" id="GLOBAL-ONEDEFTEMPL" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="A type that is used in multiple translation units shall be defined in one and only one file" id="GLOBAL-ONEDEFTYPE" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="An identifier with external linkage shall have exactly one external definition" id="GLOBAL-ONEEXTERNDEF" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="A type, object or function that is used in multiple translation units shall be declared in one and only one file" id="GLOBAL-ONEFILEDECL" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="A project shall not contain non-volatile POD variables having only one use" id="GLOBAL-ONEUSEVAR" sev="3">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Rule>

<Rule cat="GLOBAL" desc="The identifier name of a non-member object with static storage duration shall not be reused within a namespace" id="GLOBAL-REUSEDQUALGLOBVAR" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="The identifier name of a non-member static function shall not be reused within a namespace" id="GLOBAL-REUSEDQUALSTATFUN" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="All class templates, function templates, class template member functions and class template static members shall be instantiated at least one" id="GLOBAL-TEMPLNOINST" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="A class, union or enum name (including qualification, if any) shall be a unique identifier" id="GLOBAL-UNIQUETYPE" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="A typedef name (including qualification, if any) shall be a unique identifier" id="GLOBAL-UNIQUETYPEDEF" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="Every defined function with external linkage shall be used at least once" id="GLOBAL-UNUSEDFUNC" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="GLOBAL" desc="A project shall not contain unused type declarations" id="GLOBAL-UNUSEDTYPE" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="GLOBAL" desc="There shall be no unused parameters (named or unnamed) in the set of parameters for a virtual function and all the functions that override it" id="GLOBAL-UNUSEDVIRTPARAM" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="A base class shall only be declared virtual if it is used in a diamond hierarchy" id="GLOBAL-VIRTBASECLASS" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="GLOBAL" desc="Functions and objects should not be defined with external linkage if they are referenced in only one translation unit" id="GLOBAL-AVOIDEXTERN" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-10\_1\_1" desc="For multiple inheritance use virtual common base class" id="HICPP-10\_1\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-10\_2\_1" desc="Each overriding virtual function shall be declared with the override or final specifier" id="HICPP-10\_2\_1-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-10\_3\_1" desc="Be wary about using multiple inheritance of classes that are not abstract interfaces" id="HICPP-10\_3\_1-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-11\_1\_1" desc="Avoid &quot;public&quot; data members" id="HICPP-11\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-11\_1\_1" desc="Avoid 'protected' data members" id="HICPP-11\_1\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-11\_2\_1" desc="Avoid using the friend mechanism" id="HICPP-11\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_1\_1" desc="Constructors allowing for conversion should be made explicit" id="HICPP-12\_1\_1-a" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="HICPP-12\_1\_1" desc="User-conversion cast operators should be made explicit" id="HICPP-12\_1\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_2\_1" desc="Make destructors virtual in base classes" id="HICPP-12\_2\_1-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_3\_1" desc="Write operator delete if you write operator new" id="HICPP-12\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_3\_1" desc="Write operator delete[] if you write operator new[]" id="HICPP-12\_3\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_4\_1" desc="Do not use dynamic type of an object under construction" id="HICPP-12\_4\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_4\_1" desc="Do not use dynamic type of an object under destruction" id="HICPP-12\_4\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_4\_2" desc="All member variables should be initialized in constructor" id="HICPP-12\_4\_2-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_4\_3" desc="Do not specify both an NSDMI and a member initializer in a constructor for the same non-static member" id="HICPP-12\_4\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_4\_4" desc="List members in an initialization list in the order in which they are declared" id="HICPP-12\_4\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_4\_5" desc="Use delegating constructors to reduce code duplication" id="HICPP-12\_4\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_5\_1" desc="Explicitly define =default or =delete special member functions of concrete classes implicitly provided by the compiler" id="HICPP-12\_5\_1-a" sev="3">

<Stats authTot="30;" authUrg="0;" total="30"/>

</Rule>

<Rule cat="HICPP-12\_5\_2" desc="Define special members as =default when the behavior is equivalent to the compiler's behavior" id="HICPP-12\_5\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_5\_3" desc="A copy constructor shall only initialize its base classes and the non-static members of the class of which it is a member" id="HICPP-12\_5\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_5\_4" desc="All user-provided move constructors and move assignment operators shall not exit with an exception" id="HICPP-12\_5\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_5\_6" desc="User-defined copy and move assignment operators should use user-defined no-throw swap function" id="HICPP-12\_5\_6-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_5\_7" desc="Declare assignment operators with the ref-qualifier &amp;" id="HICPP-12\_5\_7-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-12\_5\_8" desc="The copy assignment operator shall be declared protected or private in an abstract class" id="HICPP-12\_5\_8-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_1\_1" desc="Write a using declaration to redeclare overloaded functions" id="HICPP-13\_1\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_1\_2" desc="Avoid Overloading on Forwarding References" id="HICPP-13\_1\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_2\_1" desc="Avoid overloading logical operators AND, OR (&amp;&amp;, ||)" id="HICPP-13\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_2\_1" desc="Avoid overloading comma operator &quot;,&quot;" id="HICPP-13\_2\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_2\_2" desc="A relational operator shall return a boolean value" id="HICPP-13\_2\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_2\_2" desc="A binary arithmetic operator and a bitwise operator shall return a 'prvalue'" id="HICPP-13\_2\_2-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_2\_3" desc="Declare binary arithmetic and bitwise operators as non-members" id="HICPP-13\_2\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_2\_4" desc="When overloading the subscript operator (operator[]), implement both const and non-const versions" id="HICPP-13\_2\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-13\_2\_5" desc="The overloaded binary operator should be implemented in terms of its corresponding compound assignment operator" id="HICPP-13\_2\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-14\_1\_1" desc="Do not use functions with variable numbers of arguments" id="HICPP-14\_1\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-14\_2\_1" desc="All partial and explicit specializations for a template shall be declared in the same file as the declaration of their primary template" id="HICPP-14\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-14\_2\_2" desc="Overloaded function templates shall not be explicitly specialized" id="HICPP-14\_2\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-14\_2\_3" desc="Declare 'extern' an explicitly instantiated template" id="HICPP-14\_2\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-15\_1\_1" desc="Only use instances of std::exception for exceptions" id="HICPP-15\_1\_1-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="HICPP-15\_1\_1" desc="Always throw the created std::exception object" id="HICPP-15\_1\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-15\_2\_1" desc="Do not throw from within destructor" id="HICPP-15\_2\_1-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-15\_3\_1" desc="Handlers of a function-try-block implementation of a class constructor or destructor shall not reference nonstatic members from this class or its bases" id="HICPP-15\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-15\_3\_2" desc="Always catch exceptions" id="HICPP-15\_3\_2-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-15\_3\_2" desc="Avoid throwing exceptions from functions that are declared not to throw" id="HICPP-15\_3\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-15\_3\_2" desc="There should be at least one exception handler to catch all otherwise unhandled exceptions" id="HICPP-15\_3\_2-b" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="Avoid using macro definitions" id="HICPP-16\_1\_1-a" sev="3">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="The #ifndef pre-processor directive will only be used to prevent multiple inclusions of the same header file" id="HICPP-16\_1\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="The #ifdef, #else, #elif preprocessor directives should not be used" id="HICPP-16\_1\_1-c" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="Use multiple include guards" id="HICPP-16\_1\_1-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="The #endif pre-processor directives will only be used to prevent multiple inclusions of the same header file" id="HICPP-16\_1\_1-e" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="The #if pre-processor directive will only be used to prevent multiple inclusions of the same header file" id="HICPP-16\_1\_1-f" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="#error directive shall not be used" id="HICPP-16\_1\_1-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="The #pragma directive shall not be used" id="HICPP-16\_1\_1-h" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_1" desc="#undef shall not be used" id="HICPP-16\_1\_1-i" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_2" desc="Do not allow absolute or relative path names in #include statements" id="HICPP-16\_1\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_2" desc="The \ character should not occur in a header file name" id="HICPP-16\_1\_2-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_3" desc="Match the filename in a #include directive to the one on the filesystem" id="HICPP-16\_1\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_4" desc="Use &lt;> brackets for system and standard library headers. Use quotes for all other headers" id="HICPP-16\_1\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-16\_1\_5" desc="Avoid duplication of #include directives" id="HICPP-16\_1\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_1\_1" desc="Avoid using vector&lt;bool>" id="HICPP-17\_1\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_2\_1" desc="The error indicator 'errno' shall not be used" id="HICPP-17\_2\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_2\_1" desc="Wrap use of the C Standard Library" id="HICPP-17\_2\_1-a" sev="4">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-17\_3\_1" desc="Do not use std::move on objects declared with the const or const &amp; type" id="HICPP-17\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_3\_2" desc="The 'std::forward' function shall be used to forward universal references" id="HICPP-17\_3\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_3\_3" desc="Do not subsequently use the argument to std::forward" id="HICPP-17\_3\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_3\_4" desc="Do not create smart pointers of array type" id="HICPP-17\_3\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_3\_5" desc="Do not create an rvalue reference of std::array" id="HICPP-17\_3\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_4\_1" desc="Use const container calls when the result is immediately converted to a const iterator" id="HICPP-17\_4\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_4\_2" desc="Prefer 'std::make\_shared' to the direct use of new" id="HICPP-17\_4\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-17\_5\_1" desc="Follow remove-like algorithms by erase to remove elements from a container" id="HICPP-17\_5\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_1\_1" desc="Do not use platform-specific multi-threading facilities" id="HICPP-18\_1\_1-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="HICPP-18\_2\_1" desc="Use high\_integrity::thread in place of std::thread" id="HICPP-18\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_2\_2" desc="Do not use global variable with different locks set" id="HICPP-18\_2\_2-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_2\_2" desc="Make const member functions thread-safe" id="HICPP-18\_2\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_2\_3" desc="Do not share volatile data between threads" id="HICPP-18\_2\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_2\_4" desc="Use std::call\_once rather than the Double-Checked Locking pattern" id="HICPP-18\_2\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_3\_1" desc="Avoid double locking" id="HICPP-18\_3\_1-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_3\_2" desc="Do not acquire locks in different order" id="HICPP-18\_3\_2-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_3\_3" desc="Do not use std::recursive\_mutex" id="HICPP-18\_3\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_3\_4" desc="Only use std::unique lock when std::lock guard cannot be used" id="HICPP-18\_3\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_3\_5" desc="Do not access the members of std::mutex directly" id="HICPP-18\_3\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_3\_6" desc="Do not use relaxed atomics" id="HICPP-18\_3\_6-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-18\_4\_1" desc="Do not use std::condition\_variable\_any on a std::mutex" id="HICPP-18\_4\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="Avoid conditions that always evaluate to the same value" id="HICPP-1\_2\_1-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="Avoid switch with unreachable branches" id="HICPP-1\_2\_1-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="There shall be no unreachable code in &quot;else&quot; block" id="HICPP-1\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="There shall be no unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="HICPP-1\_2\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="There shall be no unreachable code in &quot;if/else/while/for&quot; block" id="HICPP-1\_2\_1-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="There shall be no unreachable code in switch statement" id="HICPP-1\_2\_1-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="There shall be no unreachable code in 'for' loop" id="HICPP-1\_2\_1-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="There shall be no unreachable code after 'if' or 'switch' statement" id="HICPP-1\_2\_1-f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="There shall be no unreachable code after &quot;if&quot; or &quot;switch&quot; statement inside while/for/do...while loop" id="HICPP-1\_2\_1-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_1" desc="Avoid unreachable methods" id="HICPP-1\_2\_1-h" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_2\_2" desc="All non-null statements shall either have at least one side-effect however executed or cause control flow to change" id="HICPP-1\_2\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_3\_1" desc="Do not use the increment operator (++) on an operand of type 'bool'" id="HICPP-1\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_3\_2" desc="The 'register' storage class specifier shall not be used" id="HICPP-1\_3\_2-a" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_3\_3" desc="The C library shall not be used" id="HICPP-1\_3\_3-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-1\_3\_4" desc="Do not use deprecated STL library features" id="HICPP-1\_3\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-1\_3\_5" desc="Do not use throw exception specifications" id="HICPP-1\_3\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-2\_1\_1" desc="Tabs that do not use ASCII spaces shall not be used" id="HICPP-2\_1\_1-a" sev="5">

<Stats authTot="160;" authUrg="0;" total="160"/>

</Rule>

<Rule cat="HICPP-2\_2\_1" desc="Do not use the following digraphs: &lt;%, %>, &lt;:, :>, %:, %:%:" id="HICPP-2\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-2\_2\_1" desc="Trigraphs shall not be used" id="HICPP-2\_2\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-2\_3\_1" desc="Prefer C++ style comment" id="HICPP-2\_3\_1-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-2\_3\_2" desc="Do not use comments to remove sections of code" id="HICPP-2\_3\_2-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-2\_4\_1" desc="Different identifiers shall be typographically unambiguous" id="HICPP-2\_4\_1-a" sev="3">

<Stats authTot="13;" authUrg="0;" total="13"/>

</Rule>

<Rule cat="HICPP-2\_5\_1" desc="String literals with different encoding prefixes shall not be concatenated" id="HICPP-2\_5\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-2\_5\_2" desc="Octal constants (other than zero) shall not be used" id="HICPP-2\_5\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-2\_5\_3" desc="Prefer 'nullptr' over 'NULL' or '0'(zero)" id="HICPP-2\_5\_3-a" sev="4">

<Stats authTot="21;" authUrg="0;" total="21"/>

</Rule>

<Rule cat="HICPP-3\_1\_1" desc="Identifier declared in a local or function prototype scope shall not hide an identifier declared in a global or namespace scope" id="HICPP-3\_1\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_1\_1" desc="Identifiers declared in an inner local scope should not hide identifiers declared in an outer local scope" id="HICPP-3\_1\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_1\_1" desc="Identifiers declared in a local scope should not hide identifiers declared in a class scope" id="HICPP-3\_1\_1-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_1\_1" desc="Identifiers declared in a class scope should not hide identifiers declared in a global or namespace scope" id="HICPP-3\_1\_1-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_1\_1" desc="Identifiers declared in an inner class scope should not hide identifiers declared in outer class scope" id="HICPP-3\_1\_1-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_2\_1" desc="Always declare functions at file scope" id="HICPP-3\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_3\_1" desc="Do not use variables with static storage duration" id="HICPP-3\_3\_1-a" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="HICPP-3\_4\_1" desc="The address of an object with automatic storage shall not be returned from a function" id="HICPP-3\_4\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_4\_2" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="HICPP-3\_4\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_4\_3" desc="Prefer smart pointers over raw local pointers" id="HICPP-3\_4\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_4\_3" desc="Do not call lock() directly on a mutex" id="HICPP-3\_4\_3-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_4\_3" desc="Use RAII to prevent resource leaks" id="HICPP-3\_4\_3-c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_5\_1" desc="Do not subtract two pointers that do not address elements of the same array" id="HICPP-3\_5\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_5\_1" desc="Do not compare two unrelated pointers" id="HICPP-3\_5\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_5\_1" desc="Unions shall not be used" id="HICPP-3\_5\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-3\_5\_1" desc="typedefs should be used in place of the basic types" id="HICPP-3\_5\_1-b" sev="3">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Rule>

<Rule cat="HICPP-4\_1\_1" desc="An identifier with array type passed as a function argument shall not decay to a pointer" id="HICPP-4\_1\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_1\_1" desc="Do not pass expression with array type to a function with pointer or array type parameter" id="HICPP-4\_1\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_2\_1" desc="Apply the U suffix to literals used in a context that requires an unsigned integral expression" id="HICPP-4\_2\_1-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-4\_2\_2" desc="Avoid incorrect shift operations" id="HICPP-4\_2\_2-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_2\_2" desc="The right-hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left-hand operand" id="HICPP-4\_2\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_2\_2" desc="Avoid explicit type conversions (casts)" id="HICPP-4\_2\_2-b" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-4\_2\_2" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="HICPP-4\_2\_2-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_2\_2" desc="Avoid integer overflows" id="HICPP-4\_2\_2-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_2\_2" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="HICPP-4\_2\_2-d" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_3\_1" desc="Avoid implicit conversions from wider to narrower floating type" id="HICPP-4\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_3\_1" desc="Avoid implicit conversions of floating point numbers from wider to narrower floating type" id="HICPP-4\_3\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-4\_4\_1" desc="Avoid implicit conversions from floating to integral type" id="HICPP-4\_4\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_1" desc="Avoid magic numbers" id="HICPP-5\_1\_1-a" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="Don't write code that depends on the order of evaluation of function arguments" id="HICPP-5\_1\_2-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="HICPP-5\_1\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="HICPP-5\_1\_2-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="HICPP-5\_1\_2-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="HICPP-5\_1\_2-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="Do not use more than one volatile between two adjacent sequence points" id="HICPP-5\_1\_2-f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="Don't write code that depends on the order of evaluation of function calls" id="HICPP-5\_1\_2-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="The increment (++) and decrement (--) operators should not be mixed with other operators in an expression" id="HICPP-5\_1\_2-h" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="The comma operator shall not be used" id="HICPP-5\_1\_2-i" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_2" desc="The result of a built-in assignment operator should not be used" id="HICPP-5\_1\_2-j" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_3" desc="Use parentheses unless all operators in the expression are the same" id="HICPP-5\_1\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_3" desc="The operands of a logical &amp;&amp; or || shall be primary-expressions" id="HICPP-5\_1\_3-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_4" desc="Do not capture variables implicitly in a lambda" id="HICPP-5\_1\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_5" desc="Include a parameter list in every lambda expression" id="HICPP-5\_1\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_6" desc="The operand of the sizeof operator shall not contain any expression which has side effects" id="HICPP-5\_1\_6-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_6" desc="Object designated by a volatile lvalue should not be accessed in the operand of the sizeof operator" id="HICPP-5\_1\_6-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_6" desc="The function call that causes the side effect shall not be the operand of the sizeof operator" id="HICPP-5\_1\_6-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_6" desc="The right-hand operand of a logical &amp;&amp; or || operator shall not contain side effects" id="HICPP-5\_1\_6-d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_6" desc="The operand of the 'typeid' operator shall not contain any expression that has side effects" id="HICPP-5\_1\_6-e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_1\_6" desc="The operand of the 'typeid' operator shall not contain a function call that causes side effects" id="HICPP-5\_1\_6-f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_2\_1" desc="Avoid null pointer dereferencing" id="HICPP-5\_2\_1-c" sev="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="HICPP-5\_2\_1" desc="Avoid accessing arrays out of bounds" id="HICPP-5\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_2\_1" desc="Avoid accessing arrays and pointers out of bounds" id="HICPP-5\_2\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_2\_2" desc="Functions shall not call themselves, either directly or indirectly" id="HICPP-5\_2\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_2\_2" desc="Do not use recursion" id="HICPP-5\_2\_2-b" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_3\_1" desc="The unary minus operator shall not be applied to an expression whose underlying type is unsigned" id="HICPP-5\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_3\_2" desc="Do not use calloc, malloc, realloc and free functions" id="HICPP-5\_3\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_3\_3" desc="Properly deallocate dynamically allocated resources" id="HICPP-5\_3\_3-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_3\_3" desc="Never provide brackets ([]) for delete when deallocating non-arrays" id="HICPP-5\_3\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_3\_3" desc="Always provide empty brackets ([]) for delete when deallocating arrays" id="HICPP-5\_3\_3-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_4\_1" desc="Prefer C++-style casts" id="HICPP-5\_4\_1-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-5\_4\_1" desc="Avoid using reinterpret\_cast" id="HICPP-5\_4\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_4\_1" desc="A cast shall not remove any 'const' or 'volatile' qualification from the type of a pointer or reference" id="HICPP-5\_4\_1-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_4\_2" desc="Do not cast an expression to an enumeration type" id="HICPP-5\_4\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_4\_3" desc="Down casting (casting from base to derived class) shall not be allowed" id="HICPP-5\_4\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_5\_1" desc="Avoid division by zero" id="HICPP-5\_5\_1-a" sev="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="HICPP-5\_6\_1" desc="Bitwise operators shall only be applied to operands of unsigned underlying type" id="HICPP-5\_6\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_7\_1" desc="Floating-point expressions shall not be tested for equality or inequality" id="HICPP-5\_7\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_7\_2" desc="A pointer to member virtual function shall only be tested for equality with null-pointer-constant" id="HICPP-5\_7\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-5\_8\_1" desc="The conditional operator should not be used as a sub-expression" id="HICPP-5\_8\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_1\_1" desc="The statement forming the body of a 'switch', 'while', 'do...while' or 'for' statement shall be a compound statement" id="HICPP-6\_1\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_1\_1" desc="'if' and 'else' should be followed by a compound statement" id="HICPP-6\_1\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_1\_2" desc="All 'if...else-if' constructs shall be terminated with an 'else' clause" id="HICPP-6\_1\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_1\_2" desc="The final clause of a switch statement shall be the default clause" id="HICPP-6\_1\_2-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_1\_3" desc="An unconditional throw or break statement shall terminate every non-empty switch-clause" id="HICPP-6\_1\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_1\_4" desc="Every switch statement will have at least two cases and a potential default" id="HICPP-6\_1\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_1\_4" desc="A switch expression shall not represent a value that is effectively Boolean" id="HICPP-6\_1\_4-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_2\_1" desc="A for-loop that loops through all elements of the container and does not use its loop-counter shall not be used" id="HICPP-6\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_2\_2" desc="There shall only be one loop counter in a 'for' loop, which shall not be modified in the 'for' loop body" id="HICPP-6\_2\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_2\_3" desc="Do not modify for loop counter within a body of the loop" id="HICPP-6\_2\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_2\_4" desc="The third clause of a 'for' statement shall be well-formed" id="HICPP-6\_2\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_3\_1" desc="A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="HICPP-6\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_3\_1" desc="The goto statement shall jump to a label declared later in the same function body" id="HICPP-6\_3\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_3\_1" desc="Any label referenced by a goto statement shall be declared in the same block, or in a block enclosing the goto statement" id="HICPP-6\_3\_1-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_3\_2" desc="All exit paths from a function, except main(), with non-void return type shall have an explicit return statement with an expression" id="HICPP-6\_3\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_4\_1" desc="Declare variables as locally as possible" id="HICPP-6\_4\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-6\_4\_1" desc="Postpone variable definitions as long as possible" id="HICPP-6\_4\_1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_1\_1" desc="Multiple variable declarations shall not be allowed on the same line" id="HICPP-7\_1\_1-a" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="HICPP-7\_1\_1" desc="Each variable should be declared in a separate declaration statement" id="HICPP-7\_1\_1-b" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="HICPP-7\_1\_10" desc="Use static\_assert for assertions involving compile time constants" id="HICPP-7\_1\_10-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_1\_2" desc="Declare parameters or local variable as const whenever possible" id="HICPP-7\_1\_2-a" sev="3">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Rule>

<Rule cat="HICPP-7\_1\_3" desc="Do not place type specifiers before non-type specifiers in a declaration" id="HICPP-7\_1\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_1\_4" desc="Place CV-qualifiers on the right hand side of the type they apply to" id="HICPP-7\_1\_4-a" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="HICPP-7\_1\_5" desc="Do not inline large functions" id="HICPP-7\_1\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_1\_6" desc="The plain char type shall be used only for the storage and use of character values" id="HICPP-7\_1\_6-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_1\_6" desc="Use class types or typedefs to abstract scalar quantities and standard integer types" id="HICPP-7\_1\_6-b" sev="3">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Rule>

<Rule cat="HICPP-7\_1\_6" desc="Use class types or typedefs to abstract scalar quantities and standard integer types" id="HICPP-7\_1\_6-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_1\_7" desc="Use a trailing return type syntax if the return type is preceded by the 'typename' keyword" id="HICPP-7\_1\_7-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_1\_8" desc="Use auto id = expr when declaring a variable that is the same type as the initializer function call" id="HICPP-7\_1\_8-a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-7\_1\_9" desc="Do not explicitly specify the return type of a lambda" id="HICPP-7\_1\_9-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_2\_1" desc="Use an explicit enumeration base and ensure that it is large enough to store all enumerators" id="HICPP-7\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_2\_2" desc="Initialize all, only the first, or none of the enumerators in an enumeration" id="HICPP-7\_2\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_3\_1" desc="using-directives shall not be used" id="HICPP-7\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_4\_1" desc="Do not use static keyword except inside functions and classes" id="HICPP-7\_4\_1-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="HICPP-7\_4\_2" desc="An inline function that is used in multiple translation units shall be defined in one and only one file" id="HICPP-7\_4\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_4\_2" desc="A function template that is used in multiple translation units shall be defined in one and only one file" id="HICPP-7\_4\_2-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_4\_2" desc="A type that is used in multiple translation units shall be defined in one and only one file" id="HICPP-7\_4\_2-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_4\_3" desc="A type, object or function that is used in multiple translation units shall be declared in one and only one file" id="HICPP-7\_4\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-7\_5\_1" desc="Do not use the asm declaration" id="HICPP-7\_5\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_1\_1" desc="The declaration should not contain more than one level of pointer indirection" id="HICPP-8\_1\_1-a" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="HICPP-8\_2\_1" desc="The identifiers used in the declaration and definition of a function shall be identical" id="HICPP-8\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_2\_2" desc="Avoid functions with more than 5 parameters" id="HICPP-8\_2\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_2\_3" desc="Pass small objects with a trivial copy constructor by value" id="HICPP-8\_2\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_2\_3" desc="Pass small objects with a trivial copy constructor by value" id="HICPP-8\_2\_3-b" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="HICPP-8\_2\_4" desc="Do not pass std::unique\_ptr by const reference" id="HICPP-8\_2\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_3\_1" desc="Follow the Cyclomatic Complexity limit of 10" id="HICPP-8\_3\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_3\_3" desc="Do not use default arguments" id="HICPP-8\_3\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_3\_4" desc="Define rvalue reference functions that have a const parameter as =delete" id="HICPP-8\_3\_4-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_4\_1" desc="Avoid use before initialization" id="HICPP-8\_4\_1-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_4\_1" desc="Do not use resources that have been freed" id="HICPP-8\_4\_1-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_4\_1" desc="All automatic variables shall have been assigned a value before being used" id="HICPP-8\_4\_1-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-8\_4\_2" desc="The initializer for an aggregate or union shall be enclosed in braces" id="HICPP-8\_4\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-9\_1\_1" desc="If a member function can be made static then it shall be made static, otherwise if it can be made const then it shall be made const" id="HICPP-9\_1\_1-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="HICPP-9\_1\_2" desc="Do not redefine an inherited virtual function with a different default parameter value" id="HICPP-9\_1\_2-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-9\_1\_3" desc="Const member functions shall not return non-const pointers or references to class-data" id="HICPP-9\_1\_3-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-9\_1\_4" desc="Public member functions shall not return non-const handles to private/protected class-data" id="HICPP-9\_1\_4-a" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="HICPP-9\_1\_5" desc="Do not introduce virtual functions in a final class" id="HICPP-9\_1\_5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="HICPP-9\_2\_1" desc="Bit-fields shall have explicitly unsigned integral or enumeration types only" id="HICPP-9\_2\_1-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Do not initialize a reference to an object whose address can be changed" id="INIT-05" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="All member variables should be initialized in constructor" id="INIT-06" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Initialize static class members" id="INIT-09" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Do not use uninitialized nonstatic member variables in base class constructors" id="INIT-13\_a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Do not use uninitialized nonstatic member variables of nonvirtual base classes in virtual base class constructors" id="INIT-13\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Initialize all pointer variables" id="INIT-04" sev="2">

<Stats authTot="10;" authUrg="0;" total="10"/>

</Rule>

<Rule cat="INIT" desc="An assignment operator shall assign all data members" id="INIT-11" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Objects with external linkage should not be initialized in a header file" id="INIT-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Do not initialize unsigned integer variables with signed constants" id="INIT-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Initialize all variables" id="INIT-03" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="INIT" desc="Make class members' initialization explicit by providing user-defined constructor" id="INIT-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="User-defined constructor must be provided to ensure the proper initialization of dynamically allocated class objects" id="INIT-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="List members in an initialization list in the order in which they are declared" id="INIT-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Avoid initialization order problems across translation units by replacing non-local static objects with local static objects" id="INIT-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Do not assume that members are initialized in any special order in constructors" id="INIT-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Do not use possibly uninitialized nonstatic member variables of base classes in base class constructors" id="INIT-13\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="The initializer for an aggregate or union shall be enclosed in braces" id="INIT-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="User-defined constructors that initialize data members with the same constant values across all constructors should initialize using NSDMI instead" id="INIT-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Consider adding constructor to initialize 'struct' member variables" id="INIT-15" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="INIT" desc="Prefer initialization to assignment in constructors" id="INIT-14" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All functions shall have a cyclomatic complexity number of 20 or less" id="JSF-003" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Follow the Essential Complexity limit of 1" id="JSF-003\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Capital 'L' shall be used instead of lowercase 'l' to indicate long" id="JSF-014" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Ensure that array indices are within the bounds of the array" id="JSF-015" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The error indicator 'errno' shall not be used" id="JSF-017" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The macro 'offsetof', in library &lt;stddef.h>, shall not be used" id="JSF-018" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="'setlocale' function shall not be used" id="JSF-019" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="&lt;locale.h> shall not be used" id="JSF-019\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The 'setjmp' macro and the 'longjmp' function shall not be used" id="JSF-020" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The standard header file &lt;setjmp.h> shall not be used" id="JSF-020\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The standard header file &lt;signal.h> shall not be used" id="JSF-021" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="JSF-021\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The input/output library &lt;stdio.h> shall not be used" id="JSF-022" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="JSF" desc="The library functions atof, atoi and atol from library &lt;stdlib.h> shall not be used" id="JSF-023" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="The library function 'abort' of &lt;stdlib.h> shall not be used" id="JSF-024" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The library function 'exit' of &lt;stdlib.h> shall not be used" id="JSF-024\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The library function 'getenv' of &lt;stdlib.h> shall not be used" id="JSF-024\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The library function 'system' of &lt;stdlib.h> shall not be used" id="JSF-024\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The time handling functions of library &lt;time.h> shall not be used" id="JSF-025" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The #ifdef, #else, #elif preprocessor directives should not be used" id="JSF-026" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="The #if preprocessor directive should not be used" id="JSF-026\_b" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="#error directive shall not be used" id="JSF-026\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The #pragma directive shall not be used" id="JSF-026\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="#undef shall not be used" id="JSF-026\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The #define pre-processor directive shall not be used to create inline macros" id="JSF-029" sev="2">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="JSF" desc="The #define pre-processor directive shall not be used to define constant values" id="JSF-030" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="The #include directive shall use the &lt;filename.h> notation to include header files" id="JSF-033" sev="2">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="JSF" desc="User-specified C++ external identifiers should differ in the first 64 characters" id="JSF-046\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="User-specified C++ internal identifiers should differ from external identifiers in the first 64 characters" id="JSF-046\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="User-specified C++ internal identifiers declared in the same scope should differ in the first 64 characters" id="JSF-046\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Identifiers for constant and enumerator values shall be lowercase" id="JSF-052" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The following character sequences shall not appear in header file names: ', \, /\*, //, or &quot; " id="JSF-053.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The statements forming the body of a 'while', 'do...while', 'switch' or 'for' statement shall always be enclosed in braces" id="JSF-059\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The statements forming the body of an 'if', 'else if' or 'else' statement shall always be enclosed in braces" id="JSF-059\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Declare both private copy constructor and copy assignment operator at the same time" id="JSF-068" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Freed memory shall not be accessed under any circumstances" id="JSF-070.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Destructor shall not be called manually" id="JSF-070.1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Public and protected methods shall not be invoked by class constructor" id="JSF-071" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A class's virtual functions shall not be invoked from any of its constructors" id="JSF-071.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A class's virtual functions shall not be invoked from its destructor" id="JSF-071.1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All member variables shall be initialized in constructor" id="JSF-071\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A class's virtual functions shall not be invoked from any of its constructors" id="JSF-071\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Uninitialized nonstatic member variables shall not be used in base class constructors" id="JSF-071\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Uninitialized nonstatic member variables of nonvirtual base classes shall not be used in virtual base class constructors" id="JSF-071\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Possibly uninitialized nonstatic member variables of base classes shall not be used in base class constructors" id="JSF-071\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Members of the initialization list shall be listed in the order in which they are declared in the class" id="JSF-075" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A copy constructor and a copy assignment operator shall be declared for classes that contain pointers to data items or nontrivial destructors" id="JSF-076" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A copy constructor shall copy all data members and bases" id="JSF-077" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The definition of a constructor shall not contain default arguments that produce a signature identical to that of the implicitly-declared copy constructor" id="JSF-077.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All base classes with a virtual function shall define a virtual destructor" id="JSF-078" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="JSF" desc="Call fclose() on pointer member in destructor if the pointer was used to open a file" id="JSF-079" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Call delete on pointer members in destructors" id="JSF-079\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The assignment operator shall handle self-assignment correctly" id="JSF-081" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="An assignment operator shall return a reference to \*this" id="JSF-082" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="An assignment operator shall assign all data members" id="JSF-083\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="An assignment operator shall assign all data bases" id="JSF-083\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Multiple inheritance shall be limited to at most 1 protected implementation" id="JSF-088" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A stateful virtual base shall be explicitly declared in each derived class that accesses it" id="JSF-088.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Multiple inheritance shall not use any public implementations" id="JSF-088\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A base class shall not be both virtual and non-virtual in the same hierarchy" id="JSF-089" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="An inherited nonvirtual function shall not be redefined in a derived class" id="JSF-094" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Member functions declared in derived class shall not hide functions declared in base classes" id="JSF-094\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="An inherited default parameter shall never be redefined" id="JSF-095" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Arrays shall not be treated polymorphically" id="JSF-096" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Arrays shall not be used in interfaces" id="JSF-097" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Neither operand of an equality operator (== or !=) shall be a pointer to a virtual member function" id="JSF-097.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Arrays shall not be used in private or protected methods" id="JSF-097\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Arrays shall not be used in global functions" id="JSF-097\_c" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="JSF" desc="Arrays shall not be passed to functions as array/pointer parameters" id="JSF-097\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Functions which are called with array argument shall not be declared with pointer parameter" id="JSF-097\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Functions shall always be declared at file scope" id="JSF-107" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Functions with variable numbers of arguments shall not be used" id="JSF-108" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A function shall not return a pointer or reference to a non-static local object" id="JSF-111" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A function shall not return a pointer or a reference to a parameter that is passed by const reference" id="JSF-111\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All exit points of value-returning functions shall be through return statements" id="JSF-114" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Functions shall not call themselves, either directly or indirectly" id="JSF-119" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Only valid C++ style comments (//) shall be used" id="JSF-126" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="Code that is not used (commented out) shall be deleted" id="JSF-127" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Identifier declared in a local or function prototype scope shall not hide an identifier declared in a global or namespace scope" id="JSF-135\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Identifiers declared in an inner local scope shall not hide identifiers declared in an outer local scope" id="JSF-135\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Identifiers declared in a local scope shall not hide identifiers declared in a class scope" id="JSF-135\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Identifiers declared in a class scope shall not hide identifiers declared in a global or namespace scope" id="JSF-135\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Identifiers declared in an inner class scope shall not hide identifiers declared in outer class scope" id="JSF-135\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Identifiers shall not simultaneously have both internal and external linkage in the same translation unit" id="JSF-138\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The static storage class specifier shall be used in definitions and declarations of objects and functions that have internal linkage" id="JSF-138\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The register storage class specifier shall not be used" id="JSF-140" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All automatic variables shall have been assigned a value before being used" id="JSF-142\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All variables shall be initialized before use" id="JSF-142\_b" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="Braces shall be used to indicate and match the structure in the non-zero initialization of arrays and structures" id="JSF-144" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="In an enumerator list, the '=' construct shall not be used to explicitly initialize members other than the first, unless all items are explicitly initialized" id="JSF-145" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The underlying bit representations of floating point numbers shall not be used in any way by the programmer" id="JSF-147" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Enumeration types shall be used instead of integer types (and constants) to select from a limited series of choices" id="JSF-148" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Octal constants (other than zero) shall not be used" id="JSF-149" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A string literal shall not be modified" id="JSF-151.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Multiple variable declarations shall not be allowed on the same line" id="JSF-152" sev="2">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="JSF" desc="Unions shall not be used" id="JSF-153" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Bit-fields shall have explicitly unsigned integral or enumeration types only" id="JSF-154" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All the members of a structure (or class) shall be named" id="JSF-156" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The right hand operand of a &amp;&amp; or || operator shall not contain side effects" id="JSF-157" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The operands of a logical &amp;&amp; or || shall be parenthesized if the operands contain binary operators" id="JSF-158" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Operators ||, &amp;&amp; shall not be overloaded" id="JSF-159" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Operator unary &amp; shall not be overloaded" id="JSF-159\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="An assignment expression shall be used only as the expression in an expression statement" id="JSF-160" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Signed and unsigned values shall not be mixed in comparison operations" id="JSF-162\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Signed and unsigned values shall not be mixed in second and third operand of conditional operator" id="JSF-162\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Signed and unsigned values shall not be mixed in arithmetic operations" id="JSF-162\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Unsigned arithmetic shall not be used" id="JSF-163" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the left-hand operand" id="JSF-164" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The left-hand operand of a right-shift operator shall not have a negative value" id="JSF-164.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The unary minus operator shall not be applied to an unsigned expression" id="JSF-165" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Don't assign the dividend of two integers to a floating-point type" id="JSF-167\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The comma operator shall not be used" id="JSF-168\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="More than 2 levels of pointer indirection shall not be used" id="JSF-170" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Relational operators shall not be applied to pointer types except where they point to the same array" id="JSF-171" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The address of an object with automatic storage shall not be assigned to an object which persists after the object has ceased to exist" id="JSF-173" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The null pointer shall not be de-referenced" id="JSF-174\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The null pointer shall not be de-referenced" id="JSF-174\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="NULL identifier shall not be used; use plain 0 instead" id="JSF-175" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Down casting (casting from base to derived class) shall not be allowed" id="JSF-178" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A pointer to a virtual base class shall not be converted to a pointer to a derived class" id="JSF-179" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Implicit conversions from wider to narrower integral type which may result in a loss of information shall not be used" id="JSF-180\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid implicit conversions from floating to integral type" id="JSF-180\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid implicit conversions from wider to narrower floating type" id="JSF-180\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid conversions of constant values to a narrower type" id="JSF-180\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid implicit conversions from signed to unsigned type" id="JSF-180\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Implicit conversions from integral to floating type which may result in a loss of information shall not be used" id="JSF-180\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Implicit conversions from integral constant to floating type which may result in a loss of information shall not be used" id="JSF-180\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Type casting from any type to or from pointers shall not be used" id="JSF-182" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Floating point numbers shall not implicitly be converted to integers" id="JSF-184\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="C-style casts (other than void casts) and functional notation casts (other than explicit constructor calls) shall not be used" id="JSF-185" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="There shall be no unreachable code in 'else' block" id="JSF-186\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There shall be no unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="JSF-186\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There shall be no unreachable code in 'if', 'else', 'while' and 'for' block" id="JSF-186\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There shall be no unreachable code in 'switch' statement" id="JSF-186\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There shall be no unreachable code in 'for' loop" id="JSF-186\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There shall be no unreachable code after 'if' or 'switch' statement" id="JSF-186\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There shall be no unreachable code after 'if' or 'switch' statement inside 'while', 'for' or 'do...while' loop" id="JSF-186\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Every defined function with internal linkage shall be used" id="JSF-186\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All non-null statements shall potentially have a side-effect" id="JSF-187" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The goto statement shall not be used" id="JSF-189" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The continue statement shall not be used" id="JSF-190" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The break statement shall not be used (except to terminate the cases of a switch statement)" id="JSF-191" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="For any iteration statement there shall be no more than one break or goto statement used for loop termination" id="JSF-191\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Every non-empty case clause in a switch statement shall be terminated with a break statement" id="JSF-193" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All switch statements that do not intend to test for every enumeration value shall contain a final default clause" id="JSF-194" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Floating point variables shall not be used as loop counters" id="JSF-197" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Numeric variables being used within a for loop for iteration counting shall not be modified in the body of the loop" id="JSF-201" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Floating point variables shall not be tested for exact equality or inequality" id="JSF-202" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Evaluation of expressions shall not lead to overflow/underflow" id="JSF-203" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="JSF-204.1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Don't write code that depends on the order of evaluation of function arguments" id="JSF-204.1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="JSF-204.1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="JSF-204.1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="JSF-204.1\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Do not use more than one volatile in one expression" id="JSF-204.1\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Don't write code that depends on the order of evaluation of function calls" id="JSF-204.1\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A single operation with side-effect shall only be used in the proper context" id="JSF-204\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A call of function with side-effect shall only be used in the proper context" id="JSF-204\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The second or third operand of a ternary operator '?:' shall not contain side effects" id="JSF-204\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The volatile keyword shall not be used" id="JSF-205" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Dynamic heap memory allocation shall not be used" id="JSF-206" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="C++ exceptions shall not be used" id="JSF-208" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="Typedefs that indicate size and signedness shall be used in place of the basic types" id="JSF-209" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Typedefs that indicate size and signedness shall be used in place of the basic types" id="JSF-209\_b" sev="2">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Rule>

<Rule cat="JSF" desc="Algorithms shall not make assumptions concerning the order of allocation of nonstatic data members separated by an access specifier" id="JSF-210.1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Algorithms shall not assume particular memory alignment" id="JSF-211" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Use parenthesis to clarify expression order if operators with precedence lower than arithmetic are used" id="JSF-213\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Assuming that non-local static objects, in separate translation units, are initialized in a special order shall not be done" id="JSF-214" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Any one function (or method) will contain no more than 200 logical source lines of code (L-SLOCs)" id="JSF-001" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Only those characters specified in the C++ basic source character set will be used" id="JSF-009" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Values of character types will be restricted to a defined and documented subset of ISO 10646-1" id="JSF-010" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Trigraphs will not be used" id="JSF-011" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The following digraphs will not be used &lt;%, %>, &lt;:, :>, %:, %:%:" id="JSF-012" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Wide string literals (for example L&quot;abc&quot;) will not be used" id="JSF-013" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="#ifndef, #define and #endif will be used to prevent multiple inclusions of the same header file" id="JSF-027" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The #ifndef pre-processor directive will only be used to prevent multiple inclusions of the same header file" id="JSF-028" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The #endif pre-processor directives will only be used to prevent multiple inclusions of the same header file" id="JSF-028\_b" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="The #define pre-processor directive will only be used as part of the technique to prevent multiple inclusions of the same header file" id="JSF-031" sev="3">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="JSF" desc="The #include pre-processor directive will only be used to include header (\*.h) files" id="JSF-032" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="JSF" desc="A header file will contain a mechanism that prevents multiple inclusions of itself" id="JSF-035" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Header files (\*.h) will not contain non-const variable definitions or function definitions" id="JSF-039\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Source lines will be kept to a length of 120 characters or less" id="JSF-041" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="JSF" desc="Each expression-statement will be on a separate line" id="JSF-042" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All indentations will be consistent" id="JSF-044" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Identifiers will not begin with the underscore character &quot;\_&quot;" id="JSF-047" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="JSF" desc="Identifiers will not differ by mixture of case, the underscore character, interchange of the similarly looking letters and numbers" id="JSF-048" sev="3">

<Stats authTot="20;" authUrg="0;" total="20"/>

</Rule>

<Rule cat="JSF" desc="Begin class, struct, union, enum, and typedef names with an uppercase letter" id="JSF-050" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All letters contained in function and variable names will be composed entirely of lowercase letters" id="JSF-051" sev="3">

<Stats authTot="19;" authUrg="0;" total="19"/>

</Rule>

<Rule cat="JSF" desc="Header files will always have a file name extension of &quot;.h&quot;" id="JSF-053" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="Implementation files will always have a file name extension of &quot;.cpp&quot;" id="JSF-054" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The public section of a class will be declared before the protected section and before the private section" id="JSF-057\_a" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="JSF" desc="The protected section of a class will be declared before the private section" id="JSF-057\_b" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="JSF" desc="When declaring functions with more than 2 parameters, the leading parenthesis and the first argument are to be written on the same line as the function name, each additional argument will be written on a separate line" id="JSF-058" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Braces (&quot;{}&quot;) which enclose a block will have nothing else on the line except comments" id="JSF-060\_a" sev="3">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Rule>

<Rule cat="JSF" desc="Braces (&quot;{}&quot;) which enclose a block will be placed in the same column" id="JSF-060\_b" sev="3">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Rule>

<Rule cat="JSF" desc="Braces (&quot;{}&quot;) which enclose a block will not have an empty line after &quot;{&quot; nor before &quot;}&quot;" id="JSF-060\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Braces (&quot;{}&quot;) which enclose a block will have nothing else on the line except comments" id="JSF-061" sev="3">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Rule>

<Rule cat="JSF" desc="The dereference operator '\*' and the address-of operator '&amp;' will be directly connected with the type-specifier" id="JSF-062" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="JSF" desc="There will be no white space following '.' or '->' operator" id="JSF-063\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There will be no white space preceding '.' or '->' operator" id="JSF-063\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There will be no space between a unary operator '!' or '~' and its operand" id="JSF-063\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There will be no space between a increment/decrement operator (++/--) and its operand" id="JSF-063\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There will be no space between a unary operator '&amp;', '\*', '+', '-' and its operand" id="JSF-063\_e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="There should be no space between an increment/decrement operator (++/--) and its operand in macro definition" id="JSF-063\_f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A member function that does not affect the state of an object will be declared const" id="JSF-069" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="JSF" desc="Initialization of nonstatic class members will be performed through the member initialization list rather than through assignment in the body of a constructor" id="JSF-074" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="When two operators are opposites (such as == and !=), both will be defined" id="JSF-085" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="When one of opposite operators (== and !=) is defined, the other should be defined too and one will be defined in terms of the other" id="JSF-085\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Namespaces will not be nested more than two levels deep" id="JSF-099" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All partial and explicit specializations for a template should be declared in the same file as the declaration of their primary template" id="JSF-104" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Functions with more than 7 arguments will not be used" id="JSF-110" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Functions will have a single exit point" id="JSF-113" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="If a function returns error information, then that error information will be tested" id="JSF-115" sev="3">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="JSF" desc="Each variable declaration will be commented" id="JSF-132\_a" sev="3">

<Stats authTot="21;" authUrg="0;" total="21"/>

</Rule>

<Rule cat="JSF" desc="Each typedef will be commented" id="JSF-132\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Each enumeration value will be commented" id="JSF-132\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Each structure member variable will be commented" id="JSF-132\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Every source file will be documented with an introductory comment that provides information on the file" id="JSF-133\_a" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="JSF" desc="Every source file will contain copyright information" id="JSF-133\_b" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="JSF" desc="External objects will not be declared in more than one file" id="JSF-139" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="External objects will not be declared in implementation files" id="JSF-139\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A class, structure, or enumeration will not be declared in the definition of its type" id="JSF-141" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Variables will not be introduced until they can be initialized with meaningful values" id="JSF-143" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid unused local variables" id="JSF-143\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Hexadecimal constants will be represented using all uppercase letters" id="JSF-150" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Numeric values in code will not be used; symbolic values will be used instead" id="JSF-151" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="JSF" desc="The sizeof operator will not be used on expressions that contain side effects" id="JSF-166" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The sizeof operator will not be used on expressions that contain side effects" id="JSF-166\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The sizeof operator will not be used on expressions that contain side effects" id="JSF-166\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The implementation of integer division shall be documented" id="JSF-167" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A typedef will be used to simplify program syntax when declaring function pointers" id="JSF-176" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Redundant explicit casts will not be used" id="JSF-181\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid explicit cast from derived to a base class" id="JSF-181\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Labels will not be used, except in switch statements" id="JSF-188" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All 'if...else if' constructs will contain either a final else clause or a comment indicating why a final else clause is not necessary" id="JSF-192" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A switch expression will not represent a Boolean value" id="JSF-195" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Every switch statement will have at least two cases and a potential default" id="JSF-196" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The initialization expression in a for loop will perform no actions other than to initialize the value of a single for loop parameter" id="JSF-198" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The increment expression in a for loop will perform no action other than to change a single loop parameter to the next value for the loop" id="JSF-199" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Null initialize or increment expressions in for loops will not be used; a while loop will be used instead" id="JSF-200" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Unencapsulated global data will be avoided" id="JSF-207" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="JSF" desc="A file should directly include only headers containing declarations and definitions needed to a compilation" id="JSF-037" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Tabs should be avoided" id="JSF-043" sev="4">

<Stats authTot="160;" authUrg="0;" total="160"/>

</Rule>

<Rule cat="JSF" desc="Public data should only be used in structs, not in classes" id="JSF-067\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Protected data should only be used in structs, not in classes" id="JSF-067\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid using the friend mechanism" id="JSF-070" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Hierarchies should be based on abstract classes" id="JSF-087" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Every nonlocal name, except main(), should be placed in some namespace" id="JSF-098" sev="4">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="JSF" desc="Don't write namespace usings in a header file or before an #include" id="JSF-100" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="In a class template with a dependent base, any name that may be found in that dependent base should be referred to using a qualifier" id="JSF-105" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A function definition should not be placed in a class specification" id="JSF-109" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Dereferenced local pointer initialized by new in function scope should not be returned" id="JSF-112" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Value returned by a function having a non-void return type that is not an overloaded operator shall always be used" id="JSF-115\_a" sev="4">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="JSF" desc="Built-in-types should be passed by value unless you are modifying them" id="JSF-116" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Arguments should be passed by reference" id="JSF-117" sev="4">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="JSF" desc="An object should be passed as const reference if the function should not change the value of the object" id="JSF-117.1" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid slicing function arguments / return value" id="JSF-117\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="The class object should be passed by reference if the class has non-static pointers and has no declared copy constructor" id="JSF-117\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object" id="JSF-118" sev="4">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="JSF" desc="Declare a type of parameter as typedef to pointer to const if the pointer is not used to modify the addressed object" id="JSF-118\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Only functions with 1 or 2 statements should be considered candidates for inline functions" id="JSF-121" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Trivial accessor and mutator functions should be inlined" id="JSF-122" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Classes which have only getters/setters (accessors/mutators) are not allowed" id="JSF-123" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Trivial forwarding functions should be inlined" id="JSF-124" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Pass and return by reference when possible" id="JSF-125\_a" sev="4">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="JSF" desc="Document functions in comments that precede function definitions" id="JSF-134" sev="4">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Rule>

<Rule cat="JSF" desc="Document functions in comments that precede function declarations" id="JSF-134\_b" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="JSF" desc="Objects should be defined at block scope if they are only accessed from within a single function" id="JSF-136\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Declarations of local variables should be at the smallest feasible scope" id="JSF-136\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="All declarations at file scope should be static where possible" id="JSF-137" sev="4">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="JSF" desc="Pointers to pointers should be avoided" id="JSF-169" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="User-defined conversion functions should be avoided" id="JSF-177" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Constructors allowing for conversion should be made explicit" id="JSF-177\_b" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="JSF" desc="Every possible measure should be taken to avoid type casting" id="JSF-183" sev="4">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="JSF" desc="Avoid using static\_cast on pointers" id="JSF-183\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Avoid using reinterpret\_cast" id="JSF-183\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Pointer arithmetic will not be used" id="JSF-215" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="JSF" desc="Consider using op= instead of stand-alone op" id="JSF-125\_b" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of blocks of code in a function" id="METRICS-03" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of function calls within function" id="METRICS-04" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Class inheritance level" id="METRICS-05" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of data member(s) per class should not exceed 15" id="METRICS-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of methods per class" id="METRICS-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of parameter(s) per method should not exceed 10" id="METRICS-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of private data member(s) per class" id="METRICS-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of private methods per class" id="METRICS-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of protected data member(s) per class" id="METRICS-11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of protected methods per class" id="METRICS-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of public data member(s) per class" id="METRICS-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Number of public methods per class" id="METRICS-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Avoid functions with more than 5 parameters" id="METRICS-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Macros should not use more than 5 parameters" id="METRICS-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Follow the Cyclomatic Complexity limit of 10" id="METRICS-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="The percentage of comment lines versus the total number of module lines should be between 20 and 60" id="METRICS-19" sev="3">

<Stats authTot="19;" authUrg="0;" total="19"/>

</Rule>

<Rule cat="METRICS" desc="Avoid too long functions (declarations and statements)" id="METRICS-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Avoid too long functions (blocks)" id="METRICS-21" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Avoid functions with over 75 lines of code" id="METRICS-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Nested block depth should not be higher than 5" id="METRICS-23" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Any one function (or method) will contain no more than 200 logical source lines of code (L-SLOCs)" id="METRICS-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Source lines will be kept to a length of 120 characters or less" id="METRICS-26" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="METRICS" desc="Functions with more than 7 parameters will not be used" id="METRICS-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Follow the Cyclomatic Complexity limit of 20" id="METRICS-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="No function should be longer than 60 lines of code" id="METRICS-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="The assertion density of the code should average to a minimum of two assertions per function" id="METRICS-31" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="METRICS" desc="All functions with more than 20 lines should contain at least 2 assertions" id="METRICS-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="A global function should not be called from more than 5 different functions" id="METRICS-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="A function should not call more than 7 different functions" id="METRICS-37" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="The number of statements within function should be in range 1 - 50" id="METRICS-38" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="The value of VOCF metric for a function should not be higher than 4" id="METRICS-39" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Statements within function should not be nested deeper than 4 levels" id="METRICS-40" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="The number of blocks of comments before and inside function to the number of statements in function should be > 0.2" id="METRICS-41" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Report the value of Halstead's delivered bugs (B) for a function" id="METRICS-43" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Avoid functions with over 50 lines" id="METRICS-01" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Avoid switch statements with many cases" id="METRICS-02" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Avoid structs, unions, or classes with more than 20 fields" id="METRICS-17" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Avoid source files that are longer than 500 lines" id="METRICS-24" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Report Cyclomatic Complexity" id="METRICS-29" sev="5">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Rule>

<Rule cat="METRICS" desc="Report Essential Complexity" id="METRICS-33" sev="5">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Rule>

<Rule cat="METRICS" desc="Follow the Essential Complexity limit of 4" id="METRICS-34" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Follow the Essential Complexity limit of 10" id="METRICS-35" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="METRICS" desc="Follow the Essential Complexity limit of 1" id="METRICS-42" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Only use characters defined in ISO C standard" id="MISRA-005" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Values of character types shall be restricted to a defined and documented subset of ISO 10646-1" id="MISRA-006" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not use wide string literals" id="MISRA-008" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="The basic types of char, int, short, long, float and double should not be used, but specific-length equivalents should be typedef'd" id="MISRA-013" sev="3">

<Stats authTot="28;" authUrg="0;" total="28"/>

</Rule>

<Rule cat="MISRA" desc="Explicitly declare 'char' type as signed or unsigned" id="MISRA-014" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="The underlying bit representations of floating point numbers shall not be used" id="MISRA-016" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="All functions shall be declared before use" id="MISRA-020" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Identifiers shall not simultaneously have both internal and external linkage in the same translation unit" id="MISRA-024" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="External object should not be declared in more than one file" id="MISRA-027" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="External objects should not be declared in implementation files" id="MISRA-027\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="The use of a tag shall agree with its declaration" id="MISRA-029" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="All automatic variables shall have been assigned a value before being used" id="MISRA-030" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="The right-hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left-hand operand" id="MISRA-038" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="The comma operator shall not be used, except in the control expression of a for loop" id="MISRA-042" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Implicit conversions from wider to narrower integral type which may result in a loss of information shall not be used" id="MISRA-043" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Avoid mixing arithmetic of different precisions in the same expression" id="MISRA-043\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Implicit conversions from integral to floating type which may result in a loss of information shall not be used" id="MISRA-043\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Implicit conversions from integral constant to floating type which may result in a loss of information shall not be used" id="MISRA-043\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Redundant explicit cast to the same type is not allowed" id="MISRA-044" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not use the volatile keyword" id="MISRA-046\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Assignment statements should not be nested within other assignment statements" id="MISRA-046\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Avoid possible integer overflow in expressions in which the result is cast to a wider integer type" id="MISRA-048\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Don't cast the dividend of two integers to a floating-point type" id="MISRA-048\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Avoid possible integer overflow in expressions in which the result is assigned to a variable of a wider integer type" id="MISRA-048\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Avoid unintentionally discarding the remainder of integer division" id="MISRA-048\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="A null statement shall only occur on a line by itself" id="MISRA-054" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not use the break statement" id="MISRA-058" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not use floating point variables as loop counters" id="MISRA-065" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not use functions with variable numbers of arguments" id="MISRA-069" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call" id="MISRA-071\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Functions shall always have visible prototype at the function call" id="MISRA-071\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Provide none or all identifiers for function arguments" id="MISRA-073" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Provide expression for return statement of non-void functions" id="MISRA-083" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Avoid expressions in return statements of void functions" id="MISRA-084" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="The #include directive shall be followed by either a &lt;filename> or &quot;filename&quot; sequence" id="MISRA-089" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Enclose in parentheses whole definition of a function-like macro" id="MISRA-096" sev="3">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRA" desc="Use only non-ambiguous forms of defined pre-processor operator" id="MISRA-100" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Pointer arithmetic should not be used" id="MISRA-101" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not use non-constant pointers to functions" id="MISRA-104" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="All the functions pointed to by a single pointer to function shall be identical in the number and type of parameters and the return type" id="MISRA-105" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="The NULL pointer shall not be dereferenced" id="MISRA-107\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="The NULL pointer shall not be dereferenced" id="MISRA-107\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="All members of structure or union should be fully specified" id="MISRA-108" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Unions shall not be used to access the sub-parts of larger data structure" id="MISRA-110" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="All the members of a structure (or class/union) shall be named" id="MISRA-113" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Standard library function names shall not be reused" id="MISRA-115" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not use 'setlocale' function" id="MISRA-121\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not include &lt;locale.h> header" id="MISRA-121\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Objects or functions with external linkage shall be declared in a header file" id="MISRA-023" sev="4">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="MISRA" desc="Provisions should be made for appropriate run-time checking" id="MISRA-004\_a" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Provisions should be made for appropriate run-time checking" id="MISRA-004\_b" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Use type suffix for numeric constants" id="MISRA-018\_a" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Use type suffix for numeric constants" id="MISRA-018\_b" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Use type suffix for numeric constants" id="MISRA-018\_c" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Use type suffix for numeric constants" id="MISRA-018\_d" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Declare objects at function scope" id="MISRA-022" sev="5">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="MISRA" desc="The 'register' storage class specifier shall not be used" id="MISRA-028" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Document integer division" id="MISRA-041" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Evaluation of constant unsigned integer expressions should not lead to wrap-around" id="MISRA-051" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA" desc="Do not use labels" id="MISRA-055" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Don't write code that depends on the order of evaluation of function arguments" id="MISRA2004-12\_2\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions between signed and unsigned integer types" id="MISRA2004-10\_1\_a" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no implicit conversions from integral to floating type" id="MISRA2004-10\_1\_b" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions of complex expressions" id="MISRA2004-10\_1\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions from wider to narrower types" id="MISRA2004-10\_1\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions of function return expressions" id="MISRA2004-10\_1\_e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions of complex expressions" id="MISRA2004-10\_1\_f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions of function arguments" id="MISRA2004-10\_1\_g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions of complex expressions" id="MISRA2004-10\_1\_i" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions from floating to integral type" id="MISRA2004-10\_2\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions from wider to narrower floating type" id="MISRA2004-10\_2\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions from narrower to wider floating type" id="MISRA2004-10\_2\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid implicit conversions of floating point numbers from wider to narrower floating type" id="MISRA2004-10\_2\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The value of a complex expression of integer type shall only be cast to a type of the same signedness that is no wider than the underlying type of the expression" id="MISRA2004-10\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The value of a complex expression of floating type should not be cast to a wider floating type" id="MISRA2004-10\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The value of a complex expression of floating type should not be cast to an integer type" id="MISRA2004-10\_4\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="If the bitwise operators ~ and &lt;&lt; are applied to an operand of underlying type unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand" id="MISRA2004-10\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Conversions shall not be performed between a pointer to a function and any type other than an integral type" id="MISRA2004-11\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Conversions shall not be performed between a pointer to object and any type other than an integral type, another pointer to object type or a pointer to void" id="MISRA2004-11\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Conversions shall not be performed between a pointer to object and any type other than an integral type, another pointer to object type or a pointer to void" id="MISRA2004-11\_2\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Conversions shall not be performed between a pointer to object and any type other than an integral type, another pointer to object type or a pointer to void" id="MISRA2004-11\_2\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A cast should not convert a pointer type to an integral type" id="MISRA2004-11\_3\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A cast should not convert an integral type to a pointer type" id="MISRA2004-11\_3\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A cast should not be performed between a pointer to object type and a different pointer to object type" id="MISRA2004-11\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A cast shall not remove any 'const' or 'volatile' qualification from the type of a pointer or reference" id="MISRA2004-11\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The comma operator shall not be used" id="MISRA2004-12\_10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The underlying bit representations of floating-point values shall not be used" id="MISRA2004-12\_12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The increment (++) and decrement (--) operators should not be mixed with other operators in an expression" id="MISRA2004-12\_13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Use parenthesis for the right-hand operand of an assignment operator when it contains an assignment expression" id="MISRA2004-12\_1\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Limited dependence should be placed on C's operator precedence rules in expressions" id="MISRA2004-12\_1\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Use parentheses unless all operators in the expression are the same" id="MISRA2004-12\_1\_e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2004-12\_2\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="MISRA2004-12\_2\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="MISRA2004-12\_2\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="MISRA2004-12\_2\_e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not use more than one volatile between two adjacent sequence points" id="MISRA2004-12\_2\_f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Don't write code that depends on the order of evaluation of function calls" id="MISRA2004-12\_2\_g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The operand of the sizeof operator shall not contain any expression which has side effects" id="MISRA2004-12\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Object designated by a volatile lvalue should not be accessed in the operand of the sizeof operator" id="MISRA2004-12\_3\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The function call that causes the side effect shall not be the operand of the sizeof operator" id="MISRA2004-12\_3\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The right-hand operand of a logical &amp;&amp; or || operator shall not contain side effects" id="MISRA2004-12\_4\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The operands of a logical &amp;&amp; or || shall be primary-expressions" id="MISRA2004-12\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The operands of logical operators (&amp;&amp;, || and !) should be effectively Boolean" id="MISRA2004-12\_6\_a" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2004" desc="Expressions that are effectively Boolean should not be used as operands to operators other than (&amp;&amp;, ||, !, =, ==, !=, ?:)" id="MISRA2004-12\_6\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Bitwise operators shall not be applied to operands whose underlying type is signed" id="MISRA2004-12\_7" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The right-hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left-hand operand" id="MISRA2004-12\_8" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The unary minus operator shall not be applied to an expression whose underlying type is unsigned" id="MISRA2004-12\_9" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Assignment operators shall not be used in expressions that yield a Boolean value" id="MISRA2004-13\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Tests of a value against zero should be made explicit, unless the operand is effectively Boolean" id="MISRA2004-13\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Floating-point expressions shall not be tested for equality or inequality" id="MISRA2004-13\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The controlling expression of a for statement shall not contain any objects of floating type" id="MISRA2004-13\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The three expressions of a for statement shall be concerned only with loop control" id="MISRA2004-13\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not modify for loop counter within a body of the loop" id="MISRA2004-13\_6" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_aa" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_ab" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_ac" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_ad" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_ae" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_af" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_ag" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_ah" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_ai" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_aj" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_ak" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_j" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_k" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_l" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_m" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_n" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_s" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_t" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_u" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_v" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_w" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_x" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_y" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Boolean operations whose results are invariant shall not be permitted" id="MISRA2004-13\_7\_z" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="All 'if...else-if' constructs shall be terminated with an 'else' clause" id="MISRA2004-14\_10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no unreachable code in &quot;else&quot; block" id="MISRA2004-14\_1\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="MISRA2004-14\_1\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no unreachable code in &quot;if/else/while/for&quot; block" id="MISRA2004-14\_1\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no unreachable code in switch statement" id="MISRA2004-14\_1\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no unreachable code in 'for' loop" id="MISRA2004-14\_1\_e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no unreachable code after 'if' or 'switch' statement" id="MISRA2004-14\_1\_f" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no unreachable code after &quot;if&quot; or &quot;switch&quot; statement inside while/for/do...while loop" id="MISRA2004-14\_1\_g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="All non-null statements shall either have at least one side-effect however executed or cause control flow to change" id="MISRA2004-14\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A null statement shall occur on a line by itself or be followed by a comment" id="MISRA2004-14\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The goto statement shall not be used" id="MISRA2004-14\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The continue statement shall not be used" id="MISRA2004-14\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="For any iteration statement there shall be at most one break statement used for loop termination" id="MISRA2004-14\_6" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A function shall have a single point of exit at the end of the function" id="MISRA2004-14\_7" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The statement forming the body of a 'switch', 'while', 'do...while' or 'for' statement shall be a compound statement" id="MISRA2004-14\_8" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="'if' and 'else' should be followed by a compound statement" id="MISRA2004-14\_9" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A break statement should be placed only at the end of switch clause" id="MISRA2004-15\_0\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A switch statement shall only contain switch labels and switch clauses, and no other code" id="MISRA2004-15\_0\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="MISRA2004-15\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="An unconditional break statement shall terminate every non-empty case clause" id="MISRA2004-15\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="An unconditional break statement shall terminate every non-empty default clause" id="MISRA2004-15\_2\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The final clause of a switch statement shall be the default clause" id="MISRA2004-15\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A switch expression shall not represent a value that is effectively Boolean" id="MISRA2004-15\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A switch expression shall not represent a value that is effectively Boolean" id="MISRA2004-15\_4\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Every switch statement shall have at least one case clause" id="MISRA2004-15\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Functions shall not be defined with a variable number of arguments" id="MISRA2004-16\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="If a function returns error information, then that error information shall be tested" id="MISRA2004-16\_10" sev="3">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="MISRA2004" desc="Functions shall not call themselves, either directly or indirectly" id="MISRA2004-16\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Identifiers shall be given for all of the parameters in a function prototype declaration" id="MISRA2004-16\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The identifiers used in the declaration and definition of a function shall be identical" id="MISRA2004-16\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Functions with no parameters shall be declared with parameter type void" id="MISRA2004-16\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The number of arguments passed to a function shall match the number of parameters" id="MISRA2004-16\_6" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object" id="MISRA2004-16\_7" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2004" desc="Declare a type of parameter as typedef to pointer to const if the pointer is not used to modify the addressed object" id="MISRA2004-16\_7\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="All exit paths from a function with non-void return type shall have an explicit return statement with an expression" id="MISRA2004-16\_8" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="All exit paths from a function, except main(), with non-void return type shall have an explicit return statement with an expression" id="MISRA2004-16\_8\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A function identifier shall only be used with either a preceding '&amp;', or with a parenthesised parameter list, which may be empty" id="MISRA2004-16\_9" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not apply arithmetic to pointers that don't address an array or array element" id="MISRA2004-17\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Pointer subtraction shall only be applied to pointers that address elements of the same array" id="MISRA2004-17\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc=">, >=, &lt;, &lt;= shall not be applied to pointer types except where they point to the same array" id="MISRA2004-17\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Array indexing shall be the only allowed form of pointer arithmetic" id="MISRA2004-17\_4" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2004" desc="The declaration of objects should contain no more than 2 levels of pointer indirection" id="MISRA2004-17\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The address of an object with automatic storage shall not be returned from a function" id="MISRA2004-17\_6\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="MISRA2004-17\_6\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="All structure and union types shall be complete at the end of a translation unit" id="MISRA2004-18\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="An object shall not be assigned to an overlapping object" id="MISRA2004-18\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="An object shall not be assigned to an overlapping object" id="MISRA2004-18\_2\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Unions shall not be used" id="MISRA2004-18\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="#include statements in a file should only be preceded by other preprocessor directives or comments" id="MISRA2004-19\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##" id="MISRA2004-19\_10" sev="3">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not use in preprocessor directives #if and #elif macros not defined in translation unit" id="MISRA2004-19\_11\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be at most one occurrence of the # or ## preprocessor operators in a single macro definition" id="MISRA2004-19\_12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The # and ## preprocessor operators should not be used" id="MISRA2004-19\_13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The defined preprocessor operator shall only be used in one of the two standard forms" id="MISRA2004-19\_14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Precautions shall be taken in order to prevent the contents of a header file being included twice" id="MISRA2004-19\_15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Preprocessing directives shall be syntactically meaningful even when excluded by the preprocessor" id="MISRA2004-19\_16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if or #ifdef directive to which they are related" id="MISRA2004-19\_17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Non-standard characters should not occur in header file names in #include directives" id="MISRA2004-19\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The #include directive shall be followed by either a &lt;filename> or &quot;filename&quot; sequence" id="MISRA2004-19\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Avoid keywords and basic types in macros" id="MISRA2004-19\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Macros shall not be #define'd or #undef'd within a block" id="MISRA2004-19\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="#undef shall not be used" id="MISRA2004-19\_6" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A function should be used in preference to a function-like macro" id="MISRA2004-19\_7" sev="3">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRA2004" desc="A function-like macro shall not be invoked without all of its arguments" id="MISRA2004-19\_8" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Arguments to a function-like macro shall not contain tokens that look like preprocessing directives" id="MISRA2004-19\_9" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The library functions atof, atoi and atol from library stdlib.h shall not be used" id="MISRA2004-20\_10" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2004" desc="The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="MISRA2004-20\_11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="MISRA2004-20\_11\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The 'getenv()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="MISRA2004-20\_11\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="MISRA2004-20\_11\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The time handling functions of library time.h shall not be used" id="MISRA2004-20\_12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not #define or #undef identifiers with names which start with underscore" id="MISRA2004-20\_1\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not redefine reserved words" id="MISRA2004-20\_1\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined (C90 code)" id="MISRA2004-20\_1\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not #define nor #undef identifier 'defined'" id="MISRA2004-20\_1\_e" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The names of standard library macros, objects and functions shall not be reused" id="MISRA2004-20\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The names of standard library macros, objects and functions shall not be reused (C90)" id="MISRA2004-20\_2\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The names of standard library macros, objects and functions shall not be reused (C99)" id="MISRA2004-20\_2\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The validity of values passed to library functions shall be checked" id="MISRA2004-20\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Dynamic heap memory allocation shall not be used" id="MISRA2004-20\_4" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2004" desc="The error indicator 'errno' shall not be used" id="MISRA2004-20\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The macro offsetof, in library stddef.h, shall not be used" id="MISRA2004-20\_6" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The setjmp macro and the longjmp function shall not be used" id="MISRA2004-20\_7" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The standard header file &lt;setjmp.h> shall not be used" id="MISRA2004-20\_7\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The standard header file &lt;signal.h> shall not be used" id="MISRA2004-20\_8" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="MISRA2004-20\_8\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The input/output library stdio.h shall not be used" id="MISRA2004-20\_9" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2004" desc="Assembly language shall be encapsulated and isolated" id="MISRA2004-2\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Source code shall only use /\* ... \*/ style comments" id="MISRA2004-2\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The character sequence /\* shall not be used within a C-style comment" id="MISRA2004-2\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="All uses of the #pragma directive shall be documented and explained" id="MISRA2004-3\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not mix bit-fields and other data within the same structure" id="MISRA2004-3\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Only those escape sequences that are defined in the ISO C standard shall be used" id="MISRA2004-4\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Trigraphs shall not be used" id="MISRA2004-4\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="User-specified C external identifiers should differ in the first 31 characters" id="MISRA2004-5\_1\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="User-specified C internal identifiers should differ from external identifiers in the first 31 characters" id="MISRA2004-5\_1\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="User-specified C internal identifiers declared in the same scope should differ in the first 31 characters" id="MISRA2004-5\_1\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Identifier declared in a local or function prototype scope shall not hide an identifier declared in a global or namespace scope" id="MISRA2004-5\_2\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Identifiers declared in an inner local scope should not hide identifiers declared in an outer local scope" id="MISRA2004-5\_2\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not reuse typedef names" id="MISRA2004-5\_3\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Do not reuse typedef names as a typedef name" id="MISRA2004-5\_3\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A tag name shall not be reused for other purpose within the program" id="MISRA2004-5\_4\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="A tag name shall not be reused to define a different tag" id="MISRA2004-5\_4\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The name of typedef should not be the same as the name of type that is typedef'd" id="MISRA2004-5\_4\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The plain char type shall be used only for the storage and use of character values" id="MISRA2004-6\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="signed and unsigned char type shall be used only for the storage and use of numeric values" id="MISRA2004-6\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="typedefs to basic types should contain some digits in their name" id="MISRA2004-6\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="typedefs should be used in place of the basic types" id="MISRA2004-6\_3\_b" sev="3">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Rule>

<Rule cat="MISRA2004" desc="Bit fields shall only be defined to be of type unsigned int or signed int" id="MISRA2004-6\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Named bit-fields with signed integer type shall have a length of more than one bit" id="MISRA2004-6\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Unnamed bit-fields with signed integer type shall have a length of more than one bit" id="MISRA2004-6\_5\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Octal constants (other than zero) shall not be used" id="MISRA2004-7\_1\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Octal escape sequences shall not be used" id="MISRA2004-7\_1\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The static storage class specifier shall be used in definitions and declarations of objects and functions that have internal linkage" id="MISRA2004-8\_11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation" id="MISRA2004-8\_12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Functions shall have prototype declarations and the prototype shall be visible at both the function definition and call" id="MISRA2004-8\_1\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Functions shall always have visible prototype at the function call" id="MISRA2004-8\_1\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Whenever a function is declared or defined, its type shall be explicitly stated" id="MISRA2004-8\_2\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Whenever an object is declared or defined, its type shall be explicitly stated" id="MISRA2004-8\_2\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Use identical types in declaration and definition" id="MISRA2004-8\_3\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Use identical types in declaration and definition" id="MISRA2004-8\_3\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="If objects or functions are declared more than once their types shall be compatible" id="MISRA2004-8\_4" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="There shall be no definitions of objects or functions in a header file" id="MISRA2004-8\_5" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Always declare functions at file scope" id="MISRA2004-8\_6" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Objects shall be defined at block scope if they are only accessed from within a single function" id="MISRA2004-8\_7" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="All automatic variables shall have been assigned a value before being used" id="MISRA2004-9\_1" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The initializer for an aggregate or union shall be enclosed in braces" id="MISRA2004-9\_2" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Arrays shall not be partially initialized" id="MISRA2004-9\_2\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="The non-zero initialization of structures requires an explicit initializer for each element" id="MISRA2004-9\_2\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="In an enumerator list, the &quot;=&quot; construct shall not be used to explicitly initialise members other than the first, unless all items are explicitly initialised" id="MISRA2004-9\_3" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Sections of code should not be &quot;commented out&quot;" id="MISRA2004-2\_4" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="No object or function identifier with static storage duration should be reused" id="MISRA2004-5\_5\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="No object or function identifier with static storage duration should be reused" id="MISRA2004-5\_5\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure and union member names" id="MISRA2004-5\_6" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="No identifier name should be reused" id="MISRA2004-5\_7" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Objects or functions with external linkage shall be declared in a header file" id="MISRA2004-8\_10" sev="4">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="MISRA2004" desc="A 'U' suffix shall be applied to all constants of unsigned type" id="MISRA2004-10\_6" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Limited dependence should be placed on C's operator precedence rules in expressions" id="MISRA2004-12\_1\_b" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="No parentheses are required for the operand of a unary operator" id="MISRA2004-12\_1\_c" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2004" desc="Limited dependence should be placed on C's operator precedence rules in expressions" id="MISRA2004-12\_1\_f" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Every defined function with internal linkage shall be used" id="MISRA2008-0\_1\_10" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Every defined function with external linkage shall be used" id="MISRA2008-0\_1\_10\_b" sev="2">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be no unused parameters (named or unnamed) in nonvirtual functions" id="MISRA2008-0\_1\_11" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be no unused parameters (named or unnamed) in the set of parameters for a virtual function and all the functions that override it" id="MISRA2008-0\_1\_12" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unreachable code in 'else' block" id="MISRA2008-0\_1\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="MISRA2008-0\_1\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unreachable code in 'if/else/while/for' block" id="MISRA2008-0\_1\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unreachable code in switch statement" id="MISRA2008-0\_1\_1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unreachable code in 'for' loop" id="MISRA2008-0\_1\_1\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unreachable code after 'if' or 'switch' statement outside 'for/while/catch' block" id="MISRA2008-0\_1\_1\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unreachable code after 'if' or 'switch' statement inside 'while/for/catch' block" id="MISRA2008-0\_1\_1\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_aa" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_m" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_n" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_o" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_p" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_q" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_r" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_rz" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_s" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_sz" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_t" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_u" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_v" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_w" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_x" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_y" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain infeasible paths" id="MISRA2008-0\_1\_2\_z" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unused variables" id="MISRA2008-0\_1\_3\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unused variables" id="MISRA2008-0\_1\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Eliminate unused private member variables" id="MISRA2008-0\_1\_3\_c" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain non-volatile POD variables having only one use" id="MISRA2008-0\_1\_4" sev="2">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Rule>

<Rule cat="MISRA2008" desc="A project shall not contain unused type declarations" id="MISRA2008-0\_1\_5" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid unused values" id="MISRA2008-0\_1\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The value returned by a function having a non-void return type that is not an overloaded operator shall always be used" id="MISRA2008-0\_1\_7" sev="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="MISRA2008" desc="All functions with void return type shall have external side effect(s)" id="MISRA2008-0\_1\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All functions with void return type shall have external side effect(s)" id="MISRA2008-0\_1\_8\_b" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="All non-null statements shall either have at least one side-effect however executed or cause control flow to change" id="MISRA2008-0\_1\_9" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An object shall not be assigned to an overlapping object" id="MISRA2008-0\_2\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An object shall not be assigned to an overlapping object" id="MISRA2008-0\_2\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="If a function generates error information, then that error information shall be tested" id="MISRA2008-0\_3\_2" sev="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="MISRA2008" desc="A base class shall only be declared virtual if it is used in a diamond hierarchy" id="MISRA2008-10\_1\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An accessible base class shall not be both virtual and non-virtual in the same hierarchy" id="MISRA2008-10\_1\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be no more than one definition of each virtual function on each path through the inheritance hierarchy" id="MISRA2008-10\_3\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Each overriding virtual function shall be declared with the virtual keyword" id="MISRA2008-10\_3\_2" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="A virtual function shall only be overridden by a pure virtual function if it is itself declared as pure virtual" id="MISRA2008-10\_3\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Member data in non-POD class types shall be private" id="MISRA2008-11\_0\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Do not use dynamic type of an object under construction" id="MISRA2008-12\_1\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All constructors that are callable with a single argument of fundamental type shall be declared explicit" id="MISRA2008-12\_1\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A copy constructor shall only initialize its base classes and the nonstatic members of the class of which it is a member" id="MISRA2008-12\_8\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The copy assignment operator shall be declared protected or private in an abstract class" id="MISRA2008-12\_8\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Do not declare non-member generic functions in associated namespaces" id="MISRA2008-14\_5\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A copy constructor shall be declared when there is a template constructor with a single parameter that is a generic parameter" id="MISRA2008-14\_5\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A copy assignment operator shall be declared when there is a template assignment operator with a parameter that is a generic parameter" id="MISRA2008-14\_5\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="In a class template with a dependent base, any name that may be found in that dependent base shall be referred to using a qualified-id or this->" id="MISRA2008-14\_6\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The function shall resolve to a function declared previously in the translation unit" id="MISRA2008-14\_6\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All class templates, function templates, class template member functions and class template static members shall be instantiated at least one." id="MISRA2008-14\_7\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All partial and explicit specializations for a template shall be declared in the same file as the declaration of their primary template" id="MISRA2008-14\_7\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Overloaded function templates shall not be explicitly specialized" id="MISRA2008-14\_8\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Control shall not be transferred into a try or catch block using a goto or a switch statement" id="MISRA2008-15\_0\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The assignment-expression of a throw statement shall not itself cause an exception to be thrown" id="MISRA2008-15\_1\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="NULL shall not be thrown explicitly" id="MISRA2008-15\_1\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An empty throw (throw;) shall only be used in the compound statement of a catch handler" id="MISRA2008-15\_1\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Exceptions shall be raised only after start-up and before termination of the program" id="MISRA2008-15\_3\_1" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="Handlers of a function-try-block implementation of a class constructor or destructor shall not reference non-static members from this class or its bases" id="MISRA2008-15\_3\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="MISRA2008-15\_3\_4\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="MISRA2008-15\_3\_4\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A class type exception shall always be caught by reference" id="MISRA2008-15\_3\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class" id="MISRA2008-15\_3\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Where multiple handlers are provided in a single try-catch statement or function-try-block, any ellipsis (catch-all) handler shall occur last" id="MISRA2008-15\_3\_7" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="If a function is declared with an exception-specification, then all declarations of the same function (in other translation units) shall be declared with the same set of type-ids" id="MISRA2008-15\_4\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A class destructor shall not exit with an exception" id="MISRA2008-15\_5\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Where a function's declaration includes an exception-specification, the function shall only be capable of throwing exceptions of the indicated type(s)" id="MISRA2008-15\_5\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid throwing exceptions from functions that are declared not to throw" id="MISRA2008-15\_5\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The execution of a function registered with 'std::atexit()' or 'std::at\_quick\_exit()' should not exit via an exception" id="MISRA2008-15\_5\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Never allow an exception to be thrown from a destructor, deallocation, and swap" id="MISRA2008-15\_5\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Do not throw from within destructor" id="MISRA2008-15\_5\_3\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="There should be at least one exception handler to catch all otherwise unhandled exceptions" id="MISRA2008-15\_5\_3\_d" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="An empty throw (throw;) shall only be used in the compound-statement of a catch handler" id="MISRA2008-15\_5\_3\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Exceptions shall be raised only after start-up and before termination of the program" id="MISRA2008-15\_5\_3\_f" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" id="MISRA2008-15\_5\_3\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Where a function's declaration includes an exception-specification, the function shall only be capable of throwing exceptions of the indicated type(s)" id="MISRA2008-15\_5\_3\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Function called in global or namespace scope shall not throw unhandled exceptions" id="MISRA2008-15\_5\_3\_i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Always catch exceptions" id="MISRA2008-15\_5\_3\_j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Properly define exit handlers" id="MISRA2008-15\_5\_3\_k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid throwing exceptions from functions that are declared not to throw" id="MISRA2008-15\_5\_3\_l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="#include directives in a file shall only be preceded by other preprocessor directives or comments" id="MISRA2008-16\_0\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Macros shall only be #define'd or #undef'd in the global namespace" id="MISRA2008-16\_0\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="#undef shall not be used" id="MISRA2008-16\_0\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Function-like macros shall not be defined" id="MISRA2008-16\_0\_4" sev="2">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRA2008" desc="Arguments to a function-like macro shall not contain tokens that look like preprocessing directives" id="MISRA2008-16\_0\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="In the definition of a function-like macro, each instance of a parameter shall be enclosed in parentheses, unless it is used as the operand of # or ##" id="MISRA2008-16\_0\_6" sev="2">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRA2008" desc="Do not use in preprocessor directives #if and #elif macros not defined in translation unit" id="MISRA2008-16\_0\_7\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="If the # token appears as the first token on a line, then it shall be immediately followed by a preprocessing token" id="MISRA2008-16\_0\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The defined preprocessor operator shall only be used in one of the two standard forms" id="MISRA2008-16\_1\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if or #ifdef directive to which they are related" id="MISRA2008-16\_1\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid using macro definitions" id="MISRA2008-16\_2\_1\_a" sev="2">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="MISRA2008" desc="The #ifndef pre-processor directive will only be used to prevent multiple inclusions of the same header file" id="MISRA2008-16\_2\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The #ifdef, #else, #elif preprocessor directives should not be used" id="MISRA2008-16\_2\_1\_c" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="The #endif pre-processor directives will only be used to prevent multiple inclusions of the same header file" id="MISRA2008-16\_2\_1\_d" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="The #if pre-processor directive will only be used to prevent multiple inclusions of the same header file" id="MISRA2008-16\_2\_1\_e" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="#error directive shall not be used" id="MISRA2008-16\_2\_1\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The #pragma directive shall not be used" id="MISRA2008-16\_2\_1\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="#undef shall not be used" id="MISRA2008-16\_2\_1\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="C++ macros shall only be used for: include guards, type qualifiers, or storage class specifiers" id="MISRA2008-16\_2\_2" sev="2">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="MISRA2008" desc="Include guards shall be provided" id="MISRA2008-16\_2\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The ', &quot;, /\* or // characters shall not occur in a header file name" id="MISRA2008-16\_2\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The #include directive shall be followed by either a &lt;filename> or &quot;filename&quot; sequence" id="MISRA2008-16\_2\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be at most one occurrence of the # or ## operators in a single macro definition" id="MISRA2008-16\_3\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Reserved identifiers, macros and functions in the standard library shall not be defined, redefined or undefined" id="MISRA2008-17\_0\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Reserved identifiers, macros and functions in the standard library shall not be defined, redefined or undefined" id="MISRA2008-17\_0\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Reserved identifiers, macros and functions in the standard library shall not be defined, redefined or undefined" id="MISRA2008-17\_0\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Reserved identifiers, macros and functions in the standard library shall not be defined, redefined or undefined" id="MISRA2008-17\_0\_1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Reserved identifiers, macros and functions in the standard library shall not be defined, redefined or undefined" id="MISRA2008-17\_0\_1\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The names of standard library macros and objects shall not be reused" id="MISRA2008-17\_0\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The names of standard library functions shall not be overridden" id="MISRA2008-17\_0\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The setjmp macro and the longjmp function shall not be used" id="MISRA2008-17\_0\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The standard header file &lt;setjmp.h> shall not be used" id="MISRA2008-17\_0\_5\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The C library shall not be used" id="MISRA2008-18\_0\_1" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="The library functions atof, atoi and atol from library &lt;cstdlib> shall not be used" id="MISRA2008-18\_0\_2" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="The library function 'abort' of &lt;stdlib.h> shall not be used" id="MISRA2008-18\_0\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The library function 'exit' of &lt;stdlib.h> shall not be used" id="MISRA2008-18\_0\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The library function 'getenv' of &lt;stdlib.h> shall not be used" id="MISRA2008-18\_0\_3\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The library function 'system' of &lt;stdlib.h> shall not be used" id="MISRA2008-18\_0\_3\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The time handling functions of library &lt;ctime> shall not be used" id="MISRA2008-18\_0\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The unbounded functions of library &lt;cstring> shall not be used" id="MISRA2008-18\_0\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The macro offsetof shall not be used" id="MISRA2008-18\_2\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Dynamic heap memory allocation shall not be used" id="MISRA2008-18\_4\_1" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="The standard header file &lt;csignal> shall not be used" id="MISRA2008-18\_7\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The signal handling facilities of &lt;csignal> shall not be used" id="MISRA2008-18\_7\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The error indicator errno shall not be used" id="MISRA2008-19\_3\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The stream input/output library &lt;cstdio> shall not be used" id="MISRA2008-27\_0\_1" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="Different identifiers shall be typographically unambiguous" id="MISRA2008-2\_10\_1" sev="2">

<Stats authTot="13;" authUrg="0;" total="13"/>

</Rule>

<Rule cat="MISRA2008" desc="Identifiers declared in an inner scope shall not hide an identifier declared in an outer scope" id="MISRA2008-2\_10\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Identifiers declared in an inner scope shall not hide an identifier declared in an outer scope" id="MISRA2008-2\_10\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A typedef name (including qualification, if any) shall be a unique identifier" id="MISRA2008-2\_10\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A class, union or enum name (including qualification, if any) shall be a unique identifier" id="MISRA2008-2\_10\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="MISRA2008-2\_10\_6\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="MISRA2008-2\_10\_6\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="If an identifier refers to a type, it shall not also refer to an object or a function in the same scope" id="MISRA2008-2\_10\_6\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The name of typedef should not be the same as the name of its basic type" id="MISRA2008-2\_10\_6\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Only those escape sequences that are defined in ISO/IEC 14882:2003 shall be used" id="MISRA2008-2\_13\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Octal constants (other than zero) shall not be used" id="MISRA2008-2\_13\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Octal escape sequences (other than &quot;\0&quot;) shall not be used" id="MISRA2008-2\_13\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A &quot;U&quot; suffix shall be applied to all octal or hexadecimal integer literals of unsigned type" id="MISRA2008-2\_13\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Literal suffixes shall be upper case" id="MISRA2008-2\_13\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Narrow and wide string literals shall not be concatenated" id="MISRA2008-2\_13\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Trigraphs shall not be used" id="MISRA2008-2\_3\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The character sequence /\* shall not be used within a C-style comment" id="MISRA2008-2\_7\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Sections of code shall not be &quot;commented out&quot; using C-style comments" id="MISRA2008-2\_7\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="It shall be possible to include any header file in multiple translation units without violating the One Definition Rule" id="MISRA2008-3\_1\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Functions shall not be declared at block scope" id="MISRA2008-3\_1\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="When an array is declared, its size shall either be stated explicitly or defined implicitly by initialization" id="MISRA2008-3\_1\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All declarations of an object or function shall have compatible types" id="MISRA2008-3\_2\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The One Definition Rule shall not be violated" id="MISRA2008-3\_2\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A type, object or function that is used in multiple translation units shall be declared in one and only one file" id="MISRA2008-3\_2\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An identifier with external linkage shall have exactly one external definition" id="MISRA2008-3\_2\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Objects or functions with external linkage shall be declared in a header file" id="MISRA2008-3\_3\_1" sev="2">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="MISRA2008" desc="If a function has internal linkage then all re-declarations shall include the static storage class specifier" id="MISRA2008-3\_3\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An identifier declared to be an object or type shall be defined in a block that minimizes its visibility" id="MISRA2008-3\_4\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An identifier declared to be an object or type shall be defined in a block that minimizes its visibility" id="MISRA2008-3\_4\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The types used for an object, a function return type, or a function parameter shall be token-for-token identical in all declarations and re-declarations" id="MISRA2008-3\_9\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The underlying bit representations of floating-point values shall not be used" id="MISRA2008-3\_9\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="NULL shall not be used as an integer value" id="MISRA2008-4\_10\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Literal zero (0) shall not be used as the null-pointer-constant" id="MISRA2008-4\_10\_2" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2008" desc="Expressions with type bool shall not be used as operands to built-in operators other than the assignment operator =, the logical operators &amp;&amp;, ||, !, the equality operators == and !=, the unary &amp; operator, and the conditional operator" id="MISRA2008-4\_5\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Expressions with type enum shall not be used as operands to built-in operators other than [ ], =, ==, !=, &lt;, &lt;=, >, >=, and the unary &amp; operator" id="MISRA2008-4\_5\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Expressions with type (plain) char and wchar\_t shall not be used as operands to built-in operators other than the assignment operator =, the equality operators == and !=, and the unary &amp; operator" id="MISRA2008-4\_5\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="If the bitwise operators ~ and &lt;&lt; are applied to an operand with an underlying type of unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand" id="MISRA2008-5\_0\_10" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The plain char type shall only be used for the storage and use of character values" id="MISRA2008-5\_0\_11" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="signed char and unsigned char type shall only be used for the storage and use of numeric values" id="MISRA2008-5\_0\_12" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The condition of an if-statement and the condition of an iteration-statement shall have type bool" id="MISRA2008-5\_0\_13" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The first operand of a conditional-operator shall have type bool" id="MISRA2008-5\_0\_14" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Array indexing shall be the only form of pointer arithmetic" id="MISRA2008-5\_0\_15" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid accessing arrays out of bounds" id="MISRA2008-5\_0\_16\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A pointer operand and any pointer resulting from pointer arithmetic using that operand shall both address elements of the same array" id="MISRA2008-5\_0\_16\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Subtraction between pointers shall only be applied to pointers that address elements of the same array" id="MISRA2008-5\_0\_17" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc=">, >=, &lt;, &lt;= shall not be applied to objects of pointer type, except where they point to the same array" id="MISRA2008-5\_0\_18" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The declaration of objects shall contain no more than two levels of pointer indirection" id="MISRA2008-5\_0\_19" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2008-5\_0\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2008-5\_0\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2008-5\_0\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2008-5\_0\_1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2008-5\_0\_1\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2008-5\_0\_1\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2008-5\_0\_1\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Non-constant operands to a binary bitwise operator shall have the same underlying type" id="MISRA2008-5\_0\_20" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Bitwise operators shall only be applied to operands of unsigned underlying type" id="MISRA2008-5\_0\_21" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A cvalue expression shall not be implicitly converted to a different underlying type" id="MISRA2008-5\_0\_3\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A cvalue expression shall not be implicitly converted to a different underlying type" id="MISRA2008-5\_0\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A cvalue expression shall not be implicitly converted to a different underlying type" id="MISRA2008-5\_0\_3\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An implicit integral conversion shall not change the signedness of the underlying type" id="MISRA2008-5\_0\_4\_a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be no implicit floating-integral conversions" id="MISRA2008-5\_0\_5\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be no implicit floating-integral conversions" id="MISRA2008-5\_0\_5\_b" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2008" desc="An implicit integral or floating-point conversion shall not reduce the size of the underlying type" id="MISRA2008-5\_0\_6\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An implicit integral or floating-point conversion shall not reduce the size of the underlying type" id="MISRA2008-5\_0\_6\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An implicit integral or floating-point conversion shall not reduce the size of the underlying type" id="MISRA2008-5\_0\_6\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be no explicit floating-integral conversions of a cvalue expression" id="MISRA2008-5\_0\_7\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be no explicit floating-integral conversions of a cvalue expression" id="MISRA2008-5\_0\_7\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An explicit integral or floating-point conversion shall not increase the size of the underlying type of a cvalue expression" id="MISRA2008-5\_0\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An explicit integral or floating-point conversion shall not increase the size of the underlying type of a cvalue expression" id="MISRA2008-5\_0\_8\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An explicit integral conversion shall not change the signedness of the underlying type of a cvalue expression" id="MISRA2008-5\_0\_9" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The right hand operand of a logical &amp;&amp; or || operator shall not contain side effects" id="MISRA2008-5\_14\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The overloaded binary operator should be implemented in terms of its corresponding compound assignment operator" id="MISRA2008-5\_17\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The comma operator shall not be used" id="MISRA2008-5\_18\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Each operand of a logical &amp;&amp; or || shall be a postfix-expression" id="MISRA2008-5\_2\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The &amp;&amp; operator and the || operator shall not be overloaded" id="MISRA2008-5\_2\_11" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The comma &quot;,&quot; operator shall not be overloaded" id="MISRA2008-5\_2\_11\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An identifier with array type passed as a function argument shall not decay to a pointer" id="MISRA2008-5\_2\_12" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A pointer to a virtual base class shall only be cast to a pointer to a derived class by means of dynamic\_cast" id="MISRA2008-5\_2\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="C-style casts (other than void casts) and functional notation casts (other than explicit constructor calls) shall not be used" id="MISRA2008-5\_2\_4" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="A cast shall not remove any const or volatile qualification from the type of a pointer or reference" id="MISRA2008-5\_2\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A cast shall not convert a pointer to a function to any other pointer type, including a pointer to function type" id="MISRA2008-5\_2\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An object with pointer type shall not be converted to an unrelated pointer type, either directly or indirectly" id="MISRA2008-5\_2\_7" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An object with integer type or pointer to void type shall not be converted to an object with pointer type" id="MISRA2008-5\_2\_8" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2008" desc="Each operand of the ! operator, the logical &amp;&amp; or the logical || operators shall have type bool" id="MISRA2008-5\_3\_1" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2008" desc="The unary minus operator shall not be applied to an expression whose underlying type is unsigned" id="MISRA2008-5\_3\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The unary &amp; operator shall not be overloaded" id="MISRA2008-5\_3\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Evaluation of the operand to the sizeof operator shall not contain side effects" id="MISRA2008-5\_3\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Evaluation of the operand to the sizeof operator shall not contain side effects" id="MISRA2008-5\_3\_4\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Evaluation of the operand to the sizeof operator shall not contain side effects" id="MISRA2008-5\_3\_4\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left hand operand" id="MISRA2008-5\_8\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Assignment operators shall not be used in sub-expressions" id="MISRA2008-6\_2\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Floating-point expressions shall not be directly or indirectly tested for equality or inequality" id="MISRA2008-6\_2\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment, provided that the first character following the null statement is a white-space character" id="MISRA2008-6\_2\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The statement forming the body of a switch, while, do while or for statement shall be a compound statement" id="MISRA2008-6\_3\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An if ( condition ) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement" id="MISRA2008-6\_4\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All if ... else if constructs shall be terminated with an else clause" id="MISRA2008-6\_4\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A switch statement shall be a well-formed switch statement" id="MISRA2008-6\_4\_3\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A switch statement shall be a well-formed switch statement" id="MISRA2008-6\_4\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A switch statement shall be a well-formed switch statement" id="MISRA2008-6\_4\_3\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A switch statement shall be a well-formed switch statement" id="MISRA2008-6\_4\_3\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A switch statement shall be a well-formed switch statement" id="MISRA2008-6\_4\_3\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A switch-label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="MISRA2008-6\_4\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An unconditional throw or break statement shall terminate every non-empty switch-clause" id="MISRA2008-6\_4\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The final clause of a switch statement shall be the default-clause" id="MISRA2008-6\_4\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The condition of a switch statement shall not have bool type" id="MISRA2008-6\_4\_7" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Every switch statement shall have at least one case-clause" id="MISRA2008-6\_4\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A for loop shall contain a single loop-counter which shall not have floating type" id="MISRA2008-6\_5\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="If loop-counter is not modified by -- or ++, then, within condition, the loop-counter shall only be used as an operand to &lt;=, &lt;, > or >=" id="MISRA2008-6\_5\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The loop-counter shall not be modified within condition or statement" id="MISRA2008-6\_5\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The loop-counter shall be modified by one of: --, ++, -=n, or +=n; where n remains constant for the duration of the loop" id="MISRA2008-6\_5\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A loop-control-variable other than the loop-counter shall not be modified within condition or expression" id="MISRA2008-6\_5\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A loop-control-variable other than the loop-counter which is modified in statement within a body of the loop shall have type bool" id="MISRA2008-6\_5\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Any label referenced by a goto statement shall be declared in the same block, or in a block enclosing the goto statement" id="MISRA2008-6\_6\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The goto statement shall jump to a label declared later in the same function body" id="MISRA2008-6\_6\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The continue statement shall only be used within a well-formed for loop" id="MISRA2008-6\_6\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="For any iteration statement there shall be no more than one break or goto statement used for loop termination" id="MISRA2008-6\_6\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A function shall have a single point of exit at the end of the function" id="MISRA2008-6\_6\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A variable which is not modified shall be const qualified" id="MISRA2008-7\_1\_1" sev="2">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Rule>

<Rule cat="MISRA2008" desc="A pointer parameter in a function shall be declared as pointer to const if the corresponding object is not modified" id="MISRA2008-7\_1\_2\_a" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2008" desc="A reference parameter in a function shall be declared as reference to const if the corresponding object is not modified" id="MISRA2008-7\_1\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Declare a type of parameter as typedef to pointer to const if the pointer is not used to modify the addressed object" id="MISRA2008-7\_1\_2\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An expression with enum underlying type shall only have values corresponding to the enumerators of the enumeration" id="MISRA2008-7\_2\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The global namespace shall only contain main, namespace declarations and extern &quot;C&quot; declarations" id="MISRA2008-7\_3\_1" sev="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="MISRA2008" desc="The identifier main shall not be used for a function other than the global function main" id="MISRA2008-7\_3\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="There shall be no unnamed namespaces in header files" id="MISRA2008-7\_3\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="using-directives shall not be used" id="MISRA2008-7\_3\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Multiple declarations for an identifier in the same namespace shall not straddle a using-declaration for that identifier" id="MISRA2008-7\_3\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="using-directives and using-declarations (excluding class scope or function scope using-declarations) shall not be used in header files" id="MISRA2008-7\_3\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Assembler instructions shall only be introduced using the asm declaration" id="MISRA2008-7\_4\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Assembly language shall be encapsulated and isolated" id="MISRA2008-7\_4\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A function shall not return a reference or a pointer to an automatic variable (including parameters), defined within the function" id="MISRA2008-7\_5\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="MISRA2008-7\_5\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="MISRA2008-7\_5\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A function shall not return a reference or a pointer to a parameter that is passed by reference or const reference" id="MISRA2008-7\_5\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An init-declarator-list or a member-declarator-list shall consist of a single init-declarator or member-declarator respectively" id="MISRA2008-8\_0\_1" sev="2">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="MISRA2008" desc="Parameters in an overriding virtual function shall either use the same default arguments as the function they override, or else shall not specify any default arguments" id="MISRA2008-8\_3\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Functions shall not be defined using the ellipsis notation" id="MISRA2008-8\_4\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The identifiers used for the parameters in a re-declaration of a function shall be identical to those in the declaration" id="MISRA2008-8\_4\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All exit paths from a function with non-void return type shall have an explicit return statement with an expression" id="MISRA2008-8\_4\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A function identifier shall either be used to call the function or it shall be preceded by '&amp;'" id="MISRA2008-8\_4\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All variables shall have a defined value before they are used" id="MISRA2008-8\_5\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Braces shall be used to indicate and match the structure in the non-zero initialization of arrays and structures" id="MISRA2008-8\_5\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Arrays shall not be partially initialized" id="MISRA2008-8\_5\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Structures shall not be partially initialized" id="MISRA2008-8\_5\_2\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="In an enumerator list, the = construct shall not be used to explicitly initialize members other than the first, unless all items are explicitly initialized" id="MISRA2008-8\_5\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="const member functions shall not return non-const pointers or references to class-data" id="MISRA2008-9\_3\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Protected member functions shall not return non-const handles to class-data" id="MISRA2008-9\_3\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Public member functions shall not return non-const handles to class-data" id="MISRA2008-9\_3\_2\_b" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="If a member function can be made static then it shall be made static, otherwise if it can be made const then it shall be made const" id="MISRA2008-9\_3\_3" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="Unions shall not be used" id="MISRA2008-9\_5\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Bit-fields shall be either bool type or an explicitly unsigned or signed integral type" id="MISRA2008-9\_6\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Bit-fields shall not have enum type" id="MISRA2008-9\_6\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Named bit-fields with signed integer type shall have a length of more than one bit" id="MISRA2008-9\_6\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Classes should not be derived from virtual bases" id="MISRA2008-10\_1\_1" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All accessible entity names within a multiple inheritance hierarchy should be unique" id="MISRA2008-10\_2\_1" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All constructors of a class should explicitly call a constructor for all of its immediate base classes and all virtual base classes" id="MISRA2008-12\_1\_2" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The viable function set for a function call should either contain no function specializations, or only contain function specializations" id="MISRA2008-14\_8\_2" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="An exception object should not have pointer type" id="MISRA2008-15\_0\_2" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="There should be at least one exception handler to catch all otherwise unhandled exceptions" id="MISRA2008-15\_3\_2" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="The \ character should not occur in a header file name" id="MISRA2008-16\_2\_5" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The # and ## operators should not be used" id="MISRA2008-16\_3\_2" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The identifier name of a non-member object or function with static storage duration should not be reused" id="MISRA2008-2\_10\_5\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The identifier name of a non-member object or function with static storage duration should not be reused" id="MISRA2008-2\_10\_5\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Digraphs should not be used" id="MISRA2008-2\_5\_1" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Sections of code should not be &quot;commented out&quot; using C++ comments" id="MISRA2008-2\_7\_3" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="typedefs that indicate size and signedness should be used in place of the basic numerical types" id="MISRA2008-3\_9\_2" sev="4">

<Stats authTot="28;" authUrg="0;" total="28"/>

</Rule>

<Rule cat="MISRA2008" desc="Limited dependence should be placed on C++ operator precedence rules in expressions" id="MISRA2008-5\_0\_2\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Limited dependence should be placed on C++ operator precedence rules in expressions" id="MISRA2008-5\_0\_2\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Limited dependence should be placed on C++ operator precedence rules in expressions" id="MISRA2008-5\_0\_2\_c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Limited dependence should be placed on C++ operator precedence rules in expressions" id="MISRA2008-5\_0\_2\_d" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Limited dependence should be placed on C++ operator precedence rules in expressions" id="MISRA2008-5\_0\_2\_e" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Limited dependence should be placed on C++ operator precedence rules in expressions" id="MISRA2008-5\_0\_2\_f" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="MISRA2008-5\_19\_1\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="MISRA2008-5\_19\_1\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="The increment (++) and decrement (--) operators should not be mixed with other operators in an expression" id="MISRA2008-5\_2\_10" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Casts from a base class to a derived class should not be performed on polymorphic types" id="MISRA2008-5\_2\_3" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="A cast should not convert a pointer type to an integral type" id="MISRA2008-5\_2\_9" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Functions should not call themselves, either directly or indirectly" id="MISRA2008-7\_5\_4" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid accessing arrays out of bounds" id="MISRA2008-0\_3\_1\_a" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid null pointer dereferencing" id="MISRA2008-0\_3\_1\_b" sev="5">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid division by zero" id="MISRA2008-0\_3\_1\_c" sev="5">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid buffer overflow due to defining incorrect format limits" id="MISRA2008-0\_3\_1\_d" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid overflow due to reading a not zero terminated string" id="MISRA2008-0\_3\_1\_e" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Do not check for null after dereferencing" id="MISRA2008-0\_3\_1\_f" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid overflow when reading from a buffer" id="MISRA2008-0\_3\_1\_g" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Avoid overflow when writing to a buffer" id="MISRA2008-0\_3\_1\_h" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Do not subtract two pointers that do not address elements of the same array" id="MISRA2008-0\_3\_1\_i" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Do not compare two unrelated pointers" id="MISRA2008-0\_3\_1\_j" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="Use of floating-point arithmetic shall be documented" id="MISRA2008-0\_4\_2" sev="5">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2008" desc="All uses of the #pragma directive shall be documented" id="MISRA2008-16\_6\_1" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2008" desc="All usage of assembler shall be documented" id="MISRA2008-7\_4\_1" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Precautions shall be taken in order to prevent the contents of a header file being included more than once" id="MISRA2012-DIR-4\_10" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Validate values passed to library functions" id="MISRA2012-DIR-4\_11" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Dynamic memory allocation shall not be used" id="MISRA2012-DIR-4\_12" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid tainted data in array indexes" id="MISRA2012-DIR-4\_14\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Protect against integer overflow/underflow from tainted data" id="MISRA2012-DIR-4\_14\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid buffer read overflow from tainted data" id="MISRA2012-DIR-4\_14\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid buffer write overflow from tainted data" id="MISRA2012-DIR-4\_14\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Protect against command injection" id="MISRA2012-DIR-4\_14\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Protect against file name injection" id="MISRA2012-DIR-4\_14\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Protect against SQL injection" id="MISRA2012-DIR-4\_14\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Prevent buffer overflows from tainted data" id="MISRA2012-DIR-4\_14\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid buffer overflow from tainted data due to defining incorrect format limits" id="MISRA2012-DIR-4\_14\_i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Protect against environment injection" id="MISRA2012-DIR-4\_14\_j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid printing tainted data on the output console" id="MISRA2012-DIR-4\_14\_k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Exclude unsanitized user input from format strings" id="MISRA2012-DIR-4\_14\_l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid accessing arrays out of bounds" id="MISRA2012-DIR-4\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid null pointer dereferencing" id="MISRA2012-DIR-4\_1\_b" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid division by zero" id="MISRA2012-DIR-4\_1\_c" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid buffer overflow due to defining incorrect format limits" id="MISRA2012-DIR-4\_1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid overflow due to reading a not zero terminated string" id="MISRA2012-DIR-4\_1\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Do not check for null after dereferencing" id="MISRA2012-DIR-4\_1\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid overflow when reading from a buffer" id="MISRA2012-DIR-4\_1\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid overflow when writing to a buffer" id="MISRA2012-DIR-4\_1\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Do not subtract two pointers that do not address elements of the same array" id="MISRA2012-DIR-4\_1\_i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Do not compare two unrelated pointers" id="MISRA2012-DIR-4\_1\_j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid integer overflows" id="MISRA2012-DIR-4\_1\_k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Assembly language shall be encapsulated and isolated" id="MISRA2012-DIR-4\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Consistently check the returned value of non-void functions" id="MISRA2012-DIR-4\_7\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Always check the returned value of non-void function" id="MISRA2012-DIR-4\_7\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="All resources obtained dynamically by means of Standard Library functions shall be explicitly released" id="MISRA2012-DIR-4\_13\_a" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Do not use resources that have been freed" id="MISRA2012-DIR-4\_13\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Do not free resources using invalid pointers" id="MISRA2012-DIR-4\_13\_c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Do not abandon unreleased locks" id="MISRA2012-DIR-4\_13\_d" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Avoid double locking" id="MISRA2012-DIR-4\_13\_e" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Do not release a lock that has not been acquired" id="MISRA2012-DIR-4\_13\_f" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="All usage of assembly language should be documented" id="MISRA2012-DIR-4\_2" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Sections of code should not be &quot;commented out&quot;" id="MISRA2012-DIR-4\_4" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Identifiers in the same name space with overlapping visibility should be typographically unambiguous" id="MISRA2012-DIR-4\_5" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="typedefs to basic types should contain some digits in their name" id="MISRA2012-DIR-4\_6\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="typedefs should be used in place of the basic types" id="MISRA2012-DIR-4\_6\_b" sev="4">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="Use typedefs from stdint.h instead of declaring your own in C99 code" id="MISRA2012-DIR-4\_6\_c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden" id="MISRA2012-DIR-4\_8" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-DIR" desc="A function should be used in preference to a function-like macro where they are interchangeable" id="MISRA2012-DIR-4\_9" sev="4">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The 'sizeof' operator shall not have an operand which is a function parameter declared as &quot;array of type&quot;" id="MISRA2012-RULE-12\_5" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A function shall not be declared implicitly" id="MISRA2012-RULE-17\_3" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="All exit paths from a function with non-void return type shall have an explicit return statement with an expression" id="MISRA2012-RULE-17\_4" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="All exit paths from a function, except main(), with non-void return type shall have an explicit return statement with an expression" id="MISRA2012-RULE-17\_4\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The declaration of an array parameter shall not contain the 'static' keyword between the [ ]" id="MISRA2012-RULE-17\_6" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An object shall not be assigned or copied to an overlapping object" id="MISRA2012-RULE-19\_1\_a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An object shall not be assigned or copied to an overlapping object" id="MISRA2012-RULE-19\_1\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An object shall not be assigned or copied to an overlapping object" id="MISRA2012-RULE-19\_1\_c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Any value passed to a function in &lt;ctype.h> shall be representable as an 'unsigned char' or be the value 'EOF'" id="MISRA2012-RULE-21\_13" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid overflow due to reading a not zero terminated string" id="MISRA2012-RULE-21\_17\_a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid overflow when writing to a buffer" id="MISRA2012-RULE-21\_17\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The 'size\_t' argument passed to any function in &lt;string.h> shall have an appropriate value" id="MISRA2012-RULE-21\_18" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The pointers returned by the Standard Library functions 'localeconv', 'getenv', 'setlocale' or, 'strerror' shall only be used as if they have pointer to const-qualified type" id="MISRA2012-RULE-21\_19\_a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Strings pointed by members of the structure 'lconv' should not be modified" id="MISRA2012-RULE-21\_19\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Pointers returned by certain Standard Library functions should not be used following a subsequent call to the same or related function" id="MISRA2012-RULE-21\_20" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not use resources that have been freed" id="MISRA2012-RULE-22\_2\_a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not free resources using invalid pointers" id="MISRA2012-RULE-22\_2\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid writing to a stream which has been opened as read only" id="MISRA2012-RULE-22\_4" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A pointer to a FILE object shall not be dereferenced" id="MISRA2012-RULE-22\_5\_a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A pointer to a FILE object shall not be dereferenced by a library function" id="MISRA2012-RULE-22\_5\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of a pointer to a FILE shall not be used after the associated stream has been closed" id="MISRA2012-RULE-22\_6" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid use before initialization" id="MISRA2012-RULE-9\_1" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An expression of essentially Boolean type should always be used where an operand is interpreted as a Boolean value" id="MISRA2012-RULE-10\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An operand of essentially Boolean type should not be used where an operand is interpreted as a numeric value" id="MISRA2012-RULE-10\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An operand of essentially character type should not be used where an operand is interpreted as a numeric value" id="MISRA2012-RULE-10\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An operand of essentially enum type should not be used in an arithmetic operation" id="MISRA2012-RULE-10\_1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Shift and bitwise operations should not be performed on operands of essentially signed or enum type" id="MISRA2012-RULE-10\_1\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An operand of essentially signed or enum type should not be used as right hand side operand to the bitwise shifting operator" id="MISRA2012-RULE-10\_1\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An operand of essentially unsigned type should not be used as the operand to the unary minus operator" id="MISRA2012-RULE-10\_1\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations" id="MISRA2012-RULE-10\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of an expression shall not be assigned to an object with a narrower essential type" id="MISRA2012-RULE-10\_3\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of an expression shall not be assigned to an object of a different essential type category" id="MISRA2012-RULE-10\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category" id="MISRA2012-RULE-10\_4\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The second and third operands of the ternary operator shall have the same essential type category" id="MISRA2012-RULE-10\_4\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of a composite expression shall not be assigned to an object with wider essential type" id="MISRA2012-RULE-10\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type" id="MISRA2012-RULE-10\_7\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="If a composite expression is used as one (second or third) operand of a conditional operator then the other operand shall not have wider essential type" id="MISRA2012-RULE-10\_7\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of a composite expression shall not be cast to a different essential type category or a wider essential type" id="MISRA2012-RULE-10\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Conversions shall not be performed between a pointer to a function and any other type" id="MISRA2012-RULE-11\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Conversions shall not be performed between a pointer to a function and any other type" id="MISRA2012-RULE-11\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Conversions shall not be performed between a pointer to an incomplete type and any other type" id="MISRA2012-RULE-11\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A cast shall not be performed between a pointer to object type and a pointer to a different object type" id="MISRA2012-RULE-11\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A cast shall not be performed between pointer to void and an arithmetic type" id="MISRA2012-RULE-11\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A cast shall not be performed between pointer to object and a non-integer arithmetic type" id="MISRA2012-RULE-11\_7" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A cast shall not remove any const or volatile qualification from the type pointed to by a pointer" id="MISRA2012-RULE-11\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The macro NULL shall be the only permitted form of integer null pointer constant" id="MISRA2012-RULE-11\_9\_a" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The macro NULL shall be the only permitted form of integer null pointer constant" id="MISRA2012-RULE-11\_9\_b" sev="2">

<Stats authTot="19;" authUrg="0;" total="19"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The right hand operand of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand" id="MISRA2012-RULE-12\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Initializer lists shall not contain persistent side effects" id="MISRA2012-RULE-13\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2012-RULE-13\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Don't write code that depends on the order of evaluation of function arguments" id="MISRA2012-RULE-13\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="MISRA2012-RULE-13\_2\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="MISRA2012-RULE-13\_2\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="MISRA2012-RULE-13\_2\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not use more than one volatile in one expression" id="MISRA2012-RULE-13\_2\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Don't write code that depends on the order of evaluation of function calls" id="MISRA2012-RULE-13\_2\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The right hand operand of a logical &amp;&amp; or || operator shall not contain persistent side effects" id="MISRA2012-RULE-13\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The operand of the sizeof operator shall not contain any expression which has potential side effects" id="MISRA2012-RULE-13\_6\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The operand of the sizeof operator shall not contain any expression which has potential side effects" id="MISRA2012-RULE-13\_6\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The operand of the sizeof operator shall not contain any expression which has potential side effects" id="MISRA2012-RULE-13\_6\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A loop counter in a 'for' loop shall not have essentially floating type" id="MISRA2012-RULE-14\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A loop counter in 'while' and 'do-while' loops shall not have essentially floating type" id="MISRA2012-RULE-14\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There shall only be one loop counter in a 'for' loop, which shall not be modified in the 'for' loop body" id="MISRA2012-RULE-14\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The first clause of a 'for' loop shall be well-formed" id="MISRA2012-RULE-14\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The second clause of a 'for' loop shall be well-formed" id="MISRA2012-RULE-14\_2\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The third clause of a 'for' statement shall be well-formed" id="MISRA2012-RULE-14\_2\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Controlling expressions shall not be invariant" id="MISRA2012-RULE-14\_3\_zc" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type" id="MISRA2012-RULE-14\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The goto statement shall jump to a label declared later in the same function" id="MISRA2012-RULE-15\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement" id="MISRA2012-RULE-15\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The body of an iteration-statement or a selection-statement shall be a compound-statement" id="MISRA2012-RULE-15\_6\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The body of an iteration-statement or a selection-statement shall be a compound-statement" id="MISRA2012-RULE-15\_6\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="All 'if ... else if' constructs shall be terminated with an 'else' statement" id="MISRA2012-RULE-15\_7" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A switch statement shall only contain switch labels and switch clauses, and no other code" id="MISRA2012-RULE-16\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="MISRA2012-RULE-16\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An unconditional break statement shall terminate every non-empty case clause" id="MISRA2012-RULE-16\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An unconditional break statement shall terminate every non-empty default clause" id="MISRA2012-RULE-16\_1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Always provide a default branch for switch statements" id="MISRA2012-RULE-16\_1\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A 'default' label shall have a statement or a comment before terminating 'break'" id="MISRA2012-RULE-16\_1\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A 'default' label, if it exists, shall appear as either the first or the last switch label of a switch statement" id="MISRA2012-RULE-16\_1\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Every switch statement shall have at least two switch-clauses" id="MISRA2012-RULE-16\_1\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="MISRA2012-RULE-16\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An unconditional break statement shall terminate every switch-clause" id="MISRA2012-RULE-16\_3\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An unconditional break statement shall terminate every switch-clause" id="MISRA2012-RULE-16\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Every 'switch' statement shall have a 'default' label" id="MISRA2012-RULE-16\_4\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A 'default' label shall have a statement or a comment before terminating 'break'" id="MISRA2012-RULE-16\_4\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A default label shall appear as either the first or the last switch label of a switch statement" id="MISRA2012-RULE-16\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Every switch statement shall have at least two switch-clauses" id="MISRA2012-RULE-16\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A switch-expression shall not have essentially Boolean type" id="MISRA2012-RULE-16\_7\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A switch-expression shall not have essentially Boolean type" id="MISRA2012-RULE-16\_7\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The features of &lt;stdarg.h> shall not be used" id="MISRA2012-RULE-17\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The features of &lt;stdarg.h> shall not be used" id="MISRA2012-RULE-17\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Functions shall not call themselves, either directly or indirectly" id="MISRA2012-RULE-17\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements" id="MISRA2012-RULE-17\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value returned by a function having non-void return type shall be used" id="MISRA2012-RULE-17\_7\_a" sev="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value returned by a function having non-void return type shall be used" id="MISRA2012-RULE-17\_7\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid accessing arrays out of bounds" id="MISRA2012-RULE-18\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid accessing arrays and pointers out of bounds" id="MISRA2012-RULE-18\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A pointer operand and any pointer resulting from pointer arithmetic using that operand shall both address elements of the same array" id="MISRA2012-RULE-18\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Subtraction between pointers shall only be applied to pointers that address elements of the same array" id="MISRA2012-RULE-18\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc=">, >=, &lt;, &lt;= shall not be applied to objects of pointer type, except where they point to the same array" id="MISRA2012-RULE-18\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The address of an object with automatic storage shall not be returned from a function" id="MISRA2012-RULE-18\_6\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="MISRA2012-RULE-18\_6\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Flexible array members shall not be declared" id="MISRA2012-RULE-18\_7" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Variable-length array types shall not be used" id="MISRA2012-RULE-18\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A program should not exceed the translation limits imposed by The Standard (c90)" id="MISRA2012-RULE-1\_1\_a\_c90" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A program should not exceed the translation limits imposed by The Standard (c99)" id="MISRA2012-RULE-1\_1\_a\_c99" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A program should not exceed the translation limits imposed by The Standard (c90)" id="MISRA2012-RULE-1\_1\_b\_c90" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A program should not exceed the translation limits imposed by The Standard (c99)" id="MISRA2012-RULE-1\_1\_b\_c99" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid division by zero" id="MISRA2012-RULE-1\_3\_a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid use before initialization" id="MISRA2012-RULE-1\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not use resources that have been freed" id="MISRA2012-RULE-1\_3\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid overflow when reading from a buffer" id="MISRA2012-RULE-1\_3\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid overflow when writing to a buffer" id="MISRA2012-RULE-1\_3\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRA2012-RULE-1\_3\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Don't write code that depends on the order of evaluation of function arguments" id="MISRA2012-RULE-1\_3\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="MISRA2012-RULE-1\_3\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="MISRA2012-RULE-1\_3\_i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="MISRA2012-RULE-1\_3\_j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not use more than one volatile in one expression" id="MISRA2012-RULE-1\_3\_k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Don't write code that depends on the order of evaluation of function calls" id="MISRA2012-RULE-1\_3\_l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A function shall not return a pointer or reference to a non-static local object" id="MISRA2012-RULE-1\_3\_m" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The address of an object with automatic storage shall not be assigned to an object which persists after the object has ceased to exist" id="MISRA2012-RULE-1\_3\_n" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The left-hand operand of a right-shift operator shall not have a negative value" id="MISRA2012-RULE-1\_3\_o" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The '\_Generic' operator should not be used" id="MISRA2012-RULE-1\_4\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The '\_Noreturn' function specifier should not be used" id="MISRA2012-RULE-1\_4\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The &lt;stdnoreturn.h> header file should not be used" id="MISRA2012-RULE-1\_4\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The '\_Atomic' type specifier and the '\_Atomic' type qualifier should not be used" id="MISRA2012-RULE-1\_4\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The facilities that are specified as being provided by &lt;stdatomic.h> should not be used" id="MISRA2012-RULE-1\_4\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The '\_Thread\_local' storage class specifier should not be used" id="MISRA2012-RULE-1\_4\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The facilities that are specified as being provided by &lt;threads.h> should not be used" id="MISRA2012-RULE-1\_4\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The '\_Alignas' alignment specifier and the '\_Alignof' operator should not be used" id="MISRA2012-RULE-1\_4\_h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The &lt;stdalign.h> header file shall not be used" id="MISRA2012-RULE-1\_4\_i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The '\_\_STDC\_WANT\_LIB\_EXT1\_\_' macro should not be defined to the value other than '0'" id="MISRA2012-RULE-1\_4\_j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The 'rsize\_t' type should not be used" id="MISRA2012-RULE-1\_4\_k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The 'errno\_t' type should not be used" id="MISRA2012-RULE-1\_4\_l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not use following macros: RSIZE\_MAX, L\_tmpnam\_s, TMP\_MAX\_S" id="MISRA2012-RULE-1\_4\_m" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not use the functions defined in Annex K of ISO/IEC 9899:2011 standard" id="MISRA2012-RULE-1\_4\_n" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A macro parameter immediately following a # operator shall not immediately be followed by a ## operator" id="MISRA2012-RULE-20\_11" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A macro parameter used as an operand to the # or ## operators, which is itself subject to further macro replacement, shall only be used as an operand to these operators" id="MISRA2012-RULE-20\_12" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A line whose first token is # shall be a valid preprocessing directive" id="MISRA2012-RULE-20\_13" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if, #ifdef or #ifndef directive to which they are related" id="MISRA2012-RULE-20\_14" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The ', &amp; or \ characters and the /\* or // character sequences shall not occur in a header file name" id="MISRA2012-RULE-20\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The ', &amp; or \ characters and the /\* or // character sequences shall not occur in a header file name" id="MISRA2012-RULE-20\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The #include directive shall be followed by either a &lt;filename> or &quot;filename&quot; sequence" id="MISRA2012-RULE-20\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A macro shall not be defined with the same name as a keyword" id="MISRA2012-RULE-20\_4\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A macro shall not be defined with the same name as a keyword" id="MISRA2012-RULE-20\_4\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Tokens that look like a preprocessing directive shall not occur within a macro argument" id="MISRA2012-RULE-20\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses" id="MISRA2012-RULE-20\_7" sev="2">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1" id="MISRA2012-RULE-20\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="All identifiers used in the controlling expression of #if or #elif preprocessing directives shall be #define'd before evaluation" id="MISRA2012-RULE-20\_9\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The Standard Library time and date functions shall not be used" id="MISRA2012-RULE-21\_10" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The standard header file &lt;tgmath.h> shall not be used" id="MISRA2012-RULE-21\_11" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The facilities that are specified as being provided by &lt;tgmath.h> should not be used" id="MISRA2012-RULE-21\_11\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The Standard Library function 'memcmp' shall not be used to compare null-terminated strings" id="MISRA2012-RULE-21\_14" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The pointer arguments to the Standard Library functions 'memcmp', 'memmove' and 'memcmp' shall be pointers to qualified or unqualified versions of compatible types" id="MISRA2012-RULE-21\_15" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The pointer arguments to the Standard Library function 'memcmp' shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type" id="MISRA2012-RULE-21\_16" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not #define or #undef identifiers with names which start with underscore" id="MISRA2012-RULE-21\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="#define and #undef shall not be used on a reserved identifier or reserved macro name (for C90 code)" id="MISRA2012-RULE-21\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="#define and #undef shall not be used on a reserved identifier or reserved macro name (for C99 code)" id="MISRA2012-RULE-21\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not #define nor #undef identifier 'defined'" id="MISRA2012-RULE-21\_1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The library function 'system' of &lt;stdlib.h> shall not be used" id="MISRA2012-RULE-21\_21" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An identifier with name which starts with underscore shall not be declared" id="MISRA2012-RULE-21\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A reserved identifier or macro name shall not be declared (for C90 code)" id="MISRA2012-RULE-21\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A reserved identifier or macro name shall not be declared (for C99 code)" id="MISRA2012-RULE-21\_2\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The memory allocation and deallocation functions of &lt;stdlib.h> shall not be used" id="MISRA2012-RULE-21\_3" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The standard header file &lt;setjmp.h> shall not be used" id="MISRA2012-RULE-21\_4\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The standard header file &lt;setjmp.h> shall not be used" id="MISRA2012-RULE-21\_4\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The standard header file &lt;signal.h> shall not be used" id="MISRA2012-RULE-21\_5\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The standard header file &lt;signal.h> shall not be used" id="MISRA2012-RULE-21\_5\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The Standard Library input/output functions shall not be used" id="MISRA2012-RULE-21\_6" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The atof, atoi, atol and atoll functions of &lt;stdlib.h> shall not be used" id="MISRA2012-RULE-21\_7" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The library function 'abort' of &lt;stdlib.h> shall not be used" id="MISRA2012-RULE-21\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The library function 'exit' of &lt;stdlib.h> shall not be used" id="MISRA2012-RULE-21\_8\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The library functions 'quick\_exit' and '\_Exit' of &lt;stdlib.h> shall not be used" id="MISRA2012-RULE-21\_8\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The library functions bsearch and qsort of &lt;stdlib.h> shall not be used" id="MISRA2012-RULE-21\_9" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="All resources obtained dynamically by means of Standard Library functions shall be explicitly released" id="MISRA2012-RULE-22\_1" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of 'errno' shall only be tested when the last function to be called was an errno-setting-function" id="MISRA2012-RULE-22\_10" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The same file shall not be opened for read and write access at the same time on different stream" id="MISRA2012-RULE-22\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The macro 'EOF' should be compared with the unmodified return value from the Standard Library function" id="MISRA2012-RULE-22\_7" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of 'errno' shall be set to zero prior to a call to an errno-setting-function" id="MISRA2012-RULE-22\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The value of 'errno' shall be tested against zero after calling an errno-setting-function" id="MISRA2012-RULE-22\_9" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There shall be no unreachable code in 'else' block" id="MISRA2012-RULE-2\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There shall be no unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="MISRA2012-RULE-2\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There shall be no unreachable code in 'if', 'else', 'while', 'for' block" id="MISRA2012-RULE-2\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There shall be no unreachable code in 'switch' statement" id="MISRA2012-RULE-2\_1\_d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There shall be no unreachable code in 'for' loop" id="MISRA2012-RULE-2\_1\_e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There shall be no unreachable code after 'if' or 'switch' statement" id="MISRA2012-RULE-2\_1\_f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There shall be no unreachable code after 'if' or 'switch' statement inside 'while'/'for'/'do...while' loop" id="MISRA2012-RULE-2\_1\_g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="All non-null statements shall either have at least one side-effect however executed or cause control flow to change" id="MISRA2012-RULE-2\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Avoid unused values" id="MISRA2012-RULE-2\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The character sequence /\* shall not be used within a C-style comment" id="MISRA2012-RULE-3\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The character sequence // shall not be used within a C-style comment" id="MISRA2012-RULE-3\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The character sequence /\* shall not be used within a C++-style comment" id="MISRA2012-RULE-3\_1\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Line-splicing shall not be used in // comments" id="MISRA2012-RULE-3\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Octal and hexadecimal escape sequences shall be terminated" id="MISRA2012-RULE-4\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="External identifiers shall be distinct" id="MISRA2012-RULE-5\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers declared in the file scope and in the same name space shall be distinct (c90)" id="MISRA2012-RULE-5\_2\_a\_c90" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers declared in the file scope and in the same name space shall be distinct (c99)" id="MISRA2012-RULE-5\_2\_a\_c99" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers declared in the same block scope and name space shall be distinct (c90)" id="MISRA2012-RULE-5\_2\_b\_c90" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers declared in the same block scope and name space shall be distinct (c99)" id="MISRA2012-RULE-5\_2\_b\_c99" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An identifier declared in an inner scope shall not hide an identifier declared in an outer scope" id="MISRA2012-RULE-5\_3\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An identifier declared in an inner scope shall not hide an identifier declared in an outer scope" id="MISRA2012-RULE-5\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The name of a macro should be distinct from the names of its parameters(c90)" id="MISRA2012-RULE-5\_4\_a\_c90" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The name of a macro should be distinct from the names of its parameters(c99)" id="MISRA2012-RULE-5\_4\_a\_c99" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The name of a macro should be distinct from the names of other macros that are currently defined(c90)" id="MISRA2012-RULE-5\_4\_b\_c90" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The name of a macro should be distinct from the names of other macros that are currently defined(c99)" id="MISRA2012-RULE-5\_4\_b\_c99" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers shall be distinct from macro names (c90)" id="MISRA2012-RULE-5\_5\_c90" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers shall be distinct from macro names (c99)" id="MISRA2012-RULE-5\_5\_c99" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A typedef name shall be a unique identifier" id="MISRA2012-RULE-5\_6\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A typedef name shall be a unique identifier" id="MISRA2012-RULE-5\_6\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A tag name shall not be reused for other purpose within the program" id="MISRA2012-RULE-5\_7\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A tag name shall not be reused to define a different tag" id="MISRA2012-RULE-5\_7\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers that define objects or functions with external linkage shall be unique" id="MISRA2012-RULE-5\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Bit-fields shall only be declared with an appropriate type" id="MISRA2012-RULE-6\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Single-bit named bit fields shall not be of a signed type" id="MISRA2012-RULE-6\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Octal constants shall not be used" id="MISRA2012-RULE-7\_1" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A 'u' or 'U' suffix shall be applied to all integer constants that are represented in an unsigned type" id="MISRA2012-RULE-7\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The lowercase character 'l' shall not be used in a literal suffix" id="MISRA2012-RULE-7\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A string literal shall not be assigned to an object unless the object's type is pointer to const-qualified char" id="MISRA2012-RULE-7\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An inline function shall be declared with the static storage class" id="MISRA2012-RULE-8\_10" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Within an enumerator list, the value of an implicitly-specified enumeration constant shall be unique" id="MISRA2012-RULE-8\_12" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The restrict type qualifier shall not be used" id="MISRA2012-RULE-8\_14" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Whenever a function is declared or defined, its type shall be explicitly stated" id="MISRA2012-RULE-8\_1\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Whenever an object is declared or defined, its type shall be explicitly stated" id="MISRA2012-RULE-8\_1\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers shall be given for all of the parameters in a function prototype declaration" id="MISRA2012-RULE-8\_2\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Function types shall have named parameters" id="MISRA2012-RULE-8\_2\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Function types shall be in prototype form" id="MISRA2012-RULE-8\_2\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="If objects or functions are declared more than once their types shall be compatible" id="MISRA2012-RULE-8\_3\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The identifiers used in the declaration and definition of a function shall be identical" id="MISRA2012-RULE-8\_3\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="All declarations of an object or function shall have compatible types" id="MISRA2012-RULE-8\_3\_c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A compatible declaration shall be visible when an object or function with external linkage is defined" id="MISRA2012-RULE-8\_4\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A compatible declaration shall be visible when an object or function with external linkage is defined" id="MISRA2012-RULE-8\_4\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An external object or function shall not have more than one non-defining declaration in translation unit" id="MISRA2012-RULE-8\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An identifier with external linkage shall have exactly one external definition" id="MISRA2012-RULE-8\_6" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage" id="MISRA2012-RULE-8\_8" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The initializer for an aggregate or union shall be enclosed in braces" id="MISRA2012-RULE-9\_2" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Arrays shall not be partially initialized" id="MISRA2012-RULE-9\_3" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An element of an object shall not be initialized more than once" id="MISRA2012-RULE-9\_4" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly" id="MISRA2012-RULE-9\_5" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The cast operation to essentially enumeration type is not allowed" id="MISRA2012-RULE-10\_5\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not cast from or to essentially Boolean type" id="MISRA2012-RULE-10\_5\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Do not use casts between essentially character types and essentially floating types" id="MISRA2012-RULE-10\_5\_c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A conversion should not be performed between a pointer to object and an integer type" id="MISRA2012-RULE-11\_4" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A conversion should not be performed from pointer to void into pointer to object" id="MISRA2012-RULE-11\_5" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Use parentheses unless all operators in the expression are the same" id="MISRA2012-RULE-12\_1\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The operands of a logical &amp;&amp; or || shall be primary-expressions" id="MISRA2012-RULE-12\_1\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Parenthesis shall be used with 'sizeof' statement" id="MISRA2012-RULE-12\_1\_c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The comma operator should not be used" id="MISRA2012-RULE-12\_3" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="MISRA2012-RULE-12\_4\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="MISRA2012-RULE-12\_4\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator" id="MISRA2012-RULE-13\_3" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The result of an assignment operator should not be used" id="MISRA2012-RULE-13\_4" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The goto statement should not be used" id="MISRA2012-RULE-15\_1" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There should be no more than one break or goto statement used to terminate any iteration statement" id="MISRA2012-RULE-15\_4" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A function should have a single point of exit at the end" id="MISRA2012-RULE-15\_5" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A function parameter should not be modified" id="MISRA2012-RULE-17\_8" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The +, -, += and -= operators should not be applied to an expression of pointer type" id="MISRA2012-RULE-18\_4" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Declarations should contain no more than two levels of pointer nesting" id="MISRA2012-RULE-18\_5" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The union keyword should not be used" id="MISRA2012-RULE-19\_2" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="#include directives should only be preceded by preprocessor directives or comments" id="MISRA2012-RULE-20\_1" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The # and ## preprocessor operators should not be used" id="MISRA2012-RULE-20\_10" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="#undef should not be used" id="MISRA2012-RULE-20\_5" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="The exception handling features of &lt;fenv.h> should not be used" id="MISRA2012-RULE-21\_12" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A function should not contain unused type declarations" id="MISRA2012-RULE-2\_3\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A source file should not contain unused type declarations" id="MISRA2012-RULE-2\_3\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A function should not contain unused local tag declarations" id="MISRA2012-RULE-2\_4\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A source file should not contain unused tag declarations" id="MISRA2012-RULE-2\_4\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A source file should not contain unused macro definitions" id="MISRA2012-RULE-2\_5" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A function should not contain unused label declarations" id="MISRA2012-RULE-2\_6" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="There should be no unused parameters in functions" id="MISRA2012-RULE-2\_7" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Trigraphs should not be used" id="MISRA2012-RULE-4\_2" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers that define objects or functions with internal linkage should be unique" id="MISRA2012-RULE-5\_9\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Identifiers that define objects or functions with internal linkage should be unique" id="MISRA2012-RULE-5\_9\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="When an array with external linkage is declared, its size should be explicitly specified" id="MISRA2012-RULE-8\_11" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object" id="MISRA2012-RULE-8\_13\_a" sev="4">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Declare a type of parameter as typedef to pointer to const if the pointer is not used to modify the addressed object" id="MISRA2012-RULE-8\_13\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="Functions and objects should not be defined with external linkage if they are referenced in only one translation unit" id="MISRA2012-RULE-8\_7" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRA2012-RULE" desc="An object should be defined at block scope if its identifier only appears in a single function" id="MISRA2012-RULE-8\_9" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Avoid accessing arrays out of bounds" id="MISRAC2012-DIR\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Avoid null pointer dereferencing" id="MISRAC2012-DIR\_4\_1-b" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Avoid division by zero" id="MISRAC2012-DIR\_4\_1-c" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Avoid buffer overflow due to defining incorrect format limits" id="MISRAC2012-DIR\_4\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Avoid overflow due to reading a not zero terminated string" id="MISRAC2012-DIR\_4\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Do not check for null after dereferencing" id="MISRAC2012-DIR\_4\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Avoid overflow when reading from a buffer" id="MISRAC2012-DIR\_4\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Avoid overflow when writing to a buffer" id="MISRAC2012-DIR\_4\_1-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Do not subtract two pointers that do not address elements of the same array" id="MISRAC2012-DIR\_4\_1-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Do not compare two unrelated pointers" id="MISRAC2012-DIR\_4\_1-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_1" desc="Avoid integer overflows" id="MISRAC2012-DIR\_4\_1-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_10" desc="Use multiple include guards" id="MISRAC2012-DIR\_4\_10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_11" desc="Validate values passed to library functions" id="MISRAC2012-DIR\_4\_11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_12" desc="Dynamic heap memory allocation shall not be used" id="MISRAC2012-DIR\_4\_12-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_13" desc="Ensure resources are freed" id="MISRAC2012-DIR\_4\_13-a" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_13" desc="Do not use resources that have been freed" id="MISRAC2012-DIR\_4\_13-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_13" desc="Do not free resources using invalid pointers" id="MISRAC2012-DIR\_4\_13-c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_13" desc="Do not abandon unreleased locks" id="MISRAC2012-DIR\_4\_13-d" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_13" desc="Avoid double locking" id="MISRAC2012-DIR\_4\_13-e" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_13" desc="Do not release a lock that has not been acquired" id="MISRAC2012-DIR\_4\_13-f" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Avoid tainted data in array indexes" id="MISRAC2012-DIR\_4\_14-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Protect against integer overflow/underflow from tainted data" id="MISRAC2012-DIR\_4\_14-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Avoid buffer read overflow from tainted data" id="MISRAC2012-DIR\_4\_14-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Avoid buffer write overflow from tainted data" id="MISRAC2012-DIR\_4\_14-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Protect against command injection" id="MISRAC2012-DIR\_4\_14-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Protect against file name injection" id="MISRAC2012-DIR\_4\_14-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Protect against SQL injection" id="MISRAC2012-DIR\_4\_14-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Prevent buffer overflows from tainted data" id="MISRAC2012-DIR\_4\_14-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Avoid buffer overflow from tainted data due to defining incorrect format limits" id="MISRAC2012-DIR\_4\_14-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Protect against environment injection" id="MISRAC2012-DIR\_4\_14-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Avoid printing tainted data on the output console" id="MISRAC2012-DIR\_4\_14-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_14" desc="Exclude unsanitized user input from format strings" id="MISRAC2012-DIR\_4\_14-l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_2" desc="All usage of assembler shall be documented" id="MISRAC2012-DIR\_4\_2-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_3" desc="Assembly language shall be encapsulated and isolated" id="MISRAC2012-DIR\_4\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_4" desc="Sections of code should not be &quot;commented out&quot;" id="MISRAC2012-DIR\_4\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_5" desc="Identifiers in the same name space with overlapping visibility should be typographically unambiguous" id="MISRAC2012-DIR\_4\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_6" desc="typedefs to basic types should contain some digits in their name" id="MISRAC2012-DIR\_4\_6-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_6" desc="typedefs should be used in place of the basic types" id="MISRAC2012-DIR\_4\_6-b" sev="4">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_6" desc="Use typedefs from stdint.h instead of declaring your own in C99 code" id="MISRAC2012-DIR\_4\_6-c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_7" desc="Consistently check the returned value of non-void functions" id="MISRAC2012-DIR\_4\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_7" desc="Always check the returned value of non-void function" id="MISRAC2012-DIR\_4\_7-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_8" desc="If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden" id="MISRAC2012-DIR\_4\_8-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-DIR\_4\_9" desc="A function should be used in preference to a function-like macro" id="MISRAC2012-DIR\_4\_9-a" sev="4">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_1" desc="An expression of essentially Boolean type should always be used where an operand is interpreted as a Boolean value" id="MISRAC2012-RULE\_10\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_1" desc="An operand of essentially Boolean type should not be used where an operand is interpreted as a numeric value" id="MISRAC2012-RULE\_10\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_1" desc="An operand of essentially character type should not be used where an operand is interpreted as a numeric value" id="MISRAC2012-RULE\_10\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_1" desc="An operand of essentially enum type should not be used in an arithmetic operation" id="MISRAC2012-RULE\_10\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_1" desc="Shift and bitwise operations should not be performed on operands of essentially signed or enum type" id="MISRAC2012-RULE\_10\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_1" desc="An operand of essentially signed or enum type should not be used as the right hand operand to the bitwise shifting operator" id="MISRAC2012-RULE\_10\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_1" desc="An operand of essentially unsigned type should not be used as the operand to the unary minus operator" id="MISRAC2012-RULE\_10\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_2" desc="Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations" id="MISRAC2012-RULE\_10\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_3" desc="The value of an expression shall not be assigned to an object with a narrower essential type" id="MISRAC2012-RULE\_10\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_3" desc="The value of an expression shall not be assigned to an object of a different essential type category" id="MISRAC2012-RULE\_10\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_4" desc="Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category" id="MISRAC2012-RULE\_10\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_4" desc="The second and third operands of the ternary operator shall have the same essential type category" id="MISRAC2012-RULE\_10\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_5" desc="The cast operation to essentially enumeration type is not allowed" id="MISRAC2012-RULE\_10\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_5" desc="Do not cast from or to essentially Boolean type" id="MISRAC2012-RULE\_10\_5-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_5" desc="Do not use casts between essentially character types and essentially floating types" id="MISRAC2012-RULE\_10\_5-c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_6" desc="The value of a composite expression shall not be assigned to an object with wider essential type" id="MISRAC2012-RULE\_10\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_7" desc="If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type" id="MISRAC2012-RULE\_10\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_7" desc="If a composite expression is used as one (second or third) operand of a conditional operator then the other operand shall not have wider essential type" id="MISRAC2012-RULE\_10\_7-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_10\_8" desc="The value of a composite expression shall not be cast to a different essential type category or a wider essential type" id="MISRAC2012-RULE\_10\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_1" desc="Conversions shall not be performed between a pointer to a function and any other type than pointer to function" id="MISRAC2012-RULE\_11\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_1" desc="Conversions shall not be performed between non compatible pointer to a function types" id="MISRAC2012-RULE\_11\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_2" desc="Conversions shall not be performed between a pointer to an incomplete type and any other type" id="MISRAC2012-RULE\_11\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_3" desc="A cast shall not be performed between a pointer to object type and a pointer to a different object type" id="MISRAC2012-RULE\_11\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_4" desc="A conversion should not be performed between a pointer to object and an integer type" id="MISRAC2012-RULE\_11\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_5" desc="A conversion should not be performed from pointer to void into pointer to object" id="MISRAC2012-RULE\_11\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_6" desc="A cast shall not be performed between pointer to void and an arithmetic type" id="MISRAC2012-RULE\_11\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_7" desc="A cast shall not be performed between pointer to object and a non-integer arithmetic type" id="MISRAC2012-RULE\_11\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_8" desc="A cast shall not remove any 'const' or 'volatile' qualification from the type of a pointer or reference" id="MISRAC2012-RULE\_11\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_9" desc="Literal zero (0) shall not be used as the null-pointer-constant" id="MISRAC2012-RULE\_11\_9-a" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_11\_9" desc="Use NULL instead of literal zero (0) as the null-pointer-constant" id="MISRAC2012-RULE\_11\_9-b" sev="2">

<Stats authTot="19;" authUrg="0;" total="19"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_12\_1" desc="Use parentheses unless all operators in the expression are the same" id="MISRAC2012-RULE\_12\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_12\_1" desc="The operands of a logical &amp;&amp; or || shall be primary-expressions" id="MISRAC2012-RULE\_12\_1-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_12\_1" desc="The operand of the 'sizeof' operator should be enclosed in parentheses" id="MISRAC2012-RULE\_12\_1-c" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_12\_2" desc="The right-hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left-hand operand" id="MISRAC2012-RULE\_12\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_12\_3" desc="The comma operator shall not be used" id="MISRAC2012-RULE\_12\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_12\_4" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="MISRAC2012-RULE\_12\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_12\_4" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="MISRAC2012-RULE\_12\_4-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_12\_5" desc="The 'sizeof' operator shall not have an operand which is a function parameter declared as &quot;array of type&quot;" id="MISRAC2012-RULE\_12\_5-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_1" desc="Initializer lists shall not contain persistent side effects" id="MISRAC2012-RULE\_13\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_2" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRAC2012-RULE\_13\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_2" desc="Don't write code that depends on the order of evaluation of function arguments" id="MISRAC2012-RULE\_13\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_2" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="MISRAC2012-RULE\_13\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_2" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="MISRAC2012-RULE\_13\_2-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_2" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="MISRAC2012-RULE\_13\_2-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_2" desc="Do not use more than one volatile between two adjacent sequence points" id="MISRAC2012-RULE\_13\_2-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_2" desc="Don't write code that depends on the order of evaluation of function calls" id="MISRAC2012-RULE\_13\_2-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_3" desc="A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects" id="MISRAC2012-RULE\_13\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_4" desc="The result of a built-in assignment operator should not be used" id="MISRAC2012-RULE\_13\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_5" desc="The right-hand operand of a logical &amp;&amp; or || operator shall not contain side effects" id="MISRAC2012-RULE\_13\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_6" desc="The operand of the sizeof operator shall not contain any expression which has side effects" id="MISRAC2012-RULE\_13\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_6" desc="Object designated by a volatile lvalue should not be accessed in the operand of the sizeof operator" id="MISRAC2012-RULE\_13\_6-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_13\_6" desc="The function call shall not be the operand of the sizeof operator" id="MISRAC2012-RULE\_13\_6-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_14\_1" desc="A loop counter in a 'for' loop shall not have essentially floating type" id="MISRAC2012-RULE\_14\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_14\_1" desc="A loop counter in 'while' and 'do-while' loops shall not have essentially floating type" id="MISRAC2012-RULE\_14\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_14\_2" desc="There shall only be one loop counter in a 'for' loop, which shall not be modified in the 'for' loop body" id="MISRAC2012-RULE\_14\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_14\_2" desc="The first clause of a 'for' loop shall be well-formed" id="MISRAC2012-RULE\_14\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_14\_2" desc="The second clause of a 'for' loop shall be well-formed" id="MISRAC2012-RULE\_14\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_14\_2" desc="The third clause of a 'for' statement shall be well-formed" id="MISRAC2012-RULE\_14\_2-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_14\_3" desc="Avoid conditions that always evaluate to the same value" id="MISRAC2012-RULE\_14\_3-ac" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_14\_4" desc="Tests of a value against zero should be made explicit, unless the operand is effectively Boolean" id="MISRAC2012-RULE\_14\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_15\_1" desc="The goto statement shall not be used" id="MISRAC2012-RULE\_15\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_15\_2" desc="The goto statement shall jump to a label declared later in the same function body" id="MISRAC2012-RULE\_15\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_15\_3" desc="Any label referenced by a goto statement shall be declared in the same block, or in a block enclosing the goto statement" id="MISRAC2012-RULE\_15\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_15\_4" desc="For any iteration statement there shall be no more than one break or goto statement used for loop termination" id="MISRAC2012-RULE\_15\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_15\_5" desc="A function shall have a single point of exit at the end of the function" id="MISRAC2012-RULE\_15\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_15\_6" desc="The statement forming the body of a 'switch', 'while', 'do...while' or 'for' statement shall be a compound statement" id="MISRAC2012-RULE\_15\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_15\_6" desc="'if' and 'else' should be followed by a compound statement" id="MISRAC2012-RULE\_15\_6-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_15\_7" desc="All 'if...else-if' constructs shall be terminated with an 'else' clause" id="MISRAC2012-RULE\_15\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_1" desc="A switch statement shall only contain switch labels and switch clauses, and no other code" id="MISRAC2012-RULE\_16\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_1" desc="A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="MISRAC2012-RULE\_16\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_1" desc="An unconditional break statement shall terminate every non-empty case clause" id="MISRAC2012-RULE\_16\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_1" desc="An unconditional break statement shall terminate every non-empty default clause" id="MISRAC2012-RULE\_16\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_1" desc="Always provide a default branch for switch statements" id="MISRAC2012-RULE\_16\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_1" desc="A 'default' label shall have a statement or a comment before terminating 'break'" id="MISRAC2012-RULE\_16\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_1" desc="A 'default' label, if it exists, shall appear as either the first or the last switch label of a switch statement" id="MISRAC2012-RULE\_16\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_1" desc="Every switch statement shall have at least two switch-clauses" id="MISRAC2012-RULE\_16\_1-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_2" desc="A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" id="MISRAC2012-RULE\_16\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_3" desc="An unconditional break statement shall terminate every non-empty case clause" id="MISRAC2012-RULE\_16\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_3" desc="An unconditional break statement shall terminate every non-empty default clause" id="MISRAC2012-RULE\_16\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_4" desc="Always provide a default branch for switch statements" id="MISRAC2012-RULE\_16\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_4" desc="A 'default' label shall have a statement or a comment before terminating 'break'" id="MISRAC2012-RULE\_16\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_5" desc="A 'default' label, if it exists, shall appear as either the first or the last switch label of a switch statement" id="MISRAC2012-RULE\_16\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_6" desc="Every switch statement shall have at least two switch-clauses" id="MISRAC2012-RULE\_16\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_7" desc="A switch expression shall not represent a value that is effectively Boolean" id="MISRAC2012-RULE\_16\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_16\_7" desc="A switch expression shall not represent a value that is effectively Boolean" id="MISRAC2012-RULE\_16\_7-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_1" desc="The identifiers va\_list, va\_arg, va\_start, va\_end, va\_copy should not be used" id="MISRAC2012-RULE\_17\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_1" desc="The identifiers va\_list, va\_arg, va\_start, va\_end should not be used" id="MISRAC2012-RULE\_17\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_2" desc="Functions shall not call themselves, either directly or indirectly" id="MISRAC2012-RULE\_17\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_3" desc="Functions shall always have visible prototype at the function call" id="MISRAC2012-RULE\_17\_3-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_4" desc="All exit paths from a function with non-void return type shall have an explicit return statement with an expression" id="MISRAC2012-RULE\_17\_4-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_4" desc="All exit paths from a function, except main(), with non-void return type shall have an explicit return statement with an expression" id="MISRAC2012-RULE\_17\_4-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_5" desc="The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements" id="MISRAC2012-RULE\_17\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_6" desc="The declaration of an array parameter shall not contain the 'static' keyword between the [ ]" id="MISRAC2012-RULE\_17\_6-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_7" desc="The value returned by a function having non-void return type shall be used" id="MISRAC2012-RULE\_17\_7-a" sev="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_7" desc="The value returned by a function having non-void return type shall be used" id="MISRAC2012-RULE\_17\_7-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_17\_8" desc="A function parameter should not be modified" id="MISRAC2012-RULE\_17\_8-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_1" desc="Avoid accessing arrays out of bounds" id="MISRAC2012-RULE\_18\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_1" desc="Avoid accessing arrays and pointers out of bounds" id="MISRAC2012-RULE\_18\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_1" desc="A pointer operand and any pointer resulting from pointer arithmetic using that operand shall both address elements of the same array" id="MISRAC2012-RULE\_18\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_2" desc="Do not subtract two pointers that do not address elements of the same array" id="MISRAC2012-RULE\_18\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_3" desc="Do not compare two unrelated pointers" id="MISRAC2012-RULE\_18\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_4" desc="The +, -, += and -= operators should not be applied to an expression of pointer type" id="MISRAC2012-RULE\_18\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_5" desc="The declaration of objects should contain no more than 2 levels of pointer indirection" id="MISRAC2012-RULE\_18\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_6" desc="The address of an object with automatic storage shall not be returned from a function" id="MISRAC2012-RULE\_18\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_6" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="MISRAC2012-RULE\_18\_6-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_7" desc="Flexible array members shall not be declared" id="MISRAC2012-RULE\_18\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_18\_8" desc="Variable-length array types shall not be used" id="MISRAC2012-RULE\_18\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_19\_1" desc="An object shall not be assigned to an overlapping object" id="MISRAC2012-RULE\_19\_1-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_19\_1" desc="An object shall not be assigned to an overlapping object" id="MISRAC2012-RULE\_19\_1-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_19\_1" desc="An object shall not be assigned or copied to an overlapping object" id="MISRAC2012-RULE\_19\_1-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_19\_2" desc="The union keyword should not be used" id="MISRAC2012-RULE\_19\_2-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_1" desc="A program should not exceed the translation limits imposed by The Standard (c90)" id="MISRAC2012-RULE\_1\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_1" desc="A program should not exceed the translation limits imposed by The Standard (c99)" id="MISRAC2012-RULE\_1\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_1" desc="A program should not exceed the translation limits imposed by The Standard (c90)" id="MISRAC2012-RULE\_1\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_1" desc="A program should not exceed the translation limits imposed by The Standard (c99)" id="MISRAC2012-RULE\_1\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Avoid division by zero" id="MISRAC2012-RULE\_1\_3-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Avoid use before initialization" id="MISRAC2012-RULE\_1\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Do not use resources that have been freed" id="MISRAC2012-RULE\_1\_3-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Avoid overflow when reading from a buffer" id="MISRAC2012-RULE\_1\_3-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Avoid overflow when writing to a buffer" id="MISRAC2012-RULE\_1\_3-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="The value of an expression shall be the same under any order of evaluation that the standard permits" id="MISRAC2012-RULE\_1\_3-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Don't write code that depends on the order of evaluation of function arguments" id="MISRAC2012-RULE\_1\_3-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Don't write code that depends on the order of evaluation of function designator and function arguments" id="MISRAC2012-RULE\_1\_3-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Don't write code that depends on the order of evaluation of expression that involves a function call" id="MISRAC2012-RULE\_1\_3-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Between sequence points an object shall have its stored value modified at most once by the evaluation of an expression" id="MISRAC2012-RULE\_1\_3-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Do not use more than one volatile between two adjacent sequence points" id="MISRAC2012-RULE\_1\_3-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="Don't write code that depends on the order of evaluation of function calls" id="MISRAC2012-RULE\_1\_3-l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="The address of an object with automatic storage shall not be returned from a function" id="MISRAC2012-RULE\_1\_3-m" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" id="MISRAC2012-RULE\_1\_3-n" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_3" desc="The left-hand operand of a right-shift operator shall not have a negative value" id="MISRAC2012-RULE\_1\_3-o" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The '\_Generic' operator should not be used" id="MISRAC2012-RULE\_1\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The '\_Noreturn' function specifier should not be used" id="MISRAC2012-RULE\_1\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The &lt;stdnoreturn.h> header file should not be used" id="MISRAC2012-RULE\_1\_4-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The '\_Atomic' type specifier and the '\_Atomic' type qualifier should not be used" id="MISRAC2012-RULE\_1\_4-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The facilities that are specified as being provided by &lt;stdatomic.h> should not be used" id="MISRAC2012-RULE\_1\_4-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The '\_Thread\_local' storage class specifier should not be used" id="MISRAC2012-RULE\_1\_4-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The facilities that are specified as being provided by &lt;threads.h> should not be used" id="MISRAC2012-RULE\_1\_4-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The '\_Alignas' alignment specifier and the '\_Alignof' operator should not be used" id="MISRAC2012-RULE\_1\_4-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The &lt;stdalign.h> header file shall not be used" id="MISRAC2012-RULE\_1\_4-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The '\_\_STDC\_WANT\_LIB\_EXT1\_\_' macro should not be defined to the value other than '0'" id="MISRAC2012-RULE\_1\_4-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The 'rsize\_t' type should not be used" id="MISRAC2012-RULE\_1\_4-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="The 'errno\_t' type should not be used" id="MISRAC2012-RULE\_1\_4-l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="Do not use following macros: RSIZE\_MAX, L\_tmpnam\_s, TMP\_MAX\_S" id="MISRAC2012-RULE\_1\_4-m" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_1\_4" desc="Do not use the functions defined in Annex K of ISO/IEC 9899:2011 standard" id="MISRAC2012-RULE\_1\_4-n" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_1" desc="#include statements in a file should only be preceded by other preprocessor directives or comments" id="MISRAC2012-RULE\_20\_1-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_10" desc="The # and ## preprocessor operators should not be used" id="MISRAC2012-RULE\_20\_10-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_11" desc="A macro parameter immediately following a # operator shall not immediately be followed by or preceded by a ## operator" id="MISRAC2012-RULE\_20\_11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_12" desc="A macro parameter used as an operand to the # or ## operators, which is itself subject to further macro replacement, shall only be used as an operand to these operators" id="MISRAC2012-RULE\_20\_12-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_13" desc="Preprocessing directives shall be syntactically meaningful even when excluded by the preprocessor" id="MISRAC2012-RULE\_20\_13-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_14" desc="All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if or #ifdef directive to which they are related" id="MISRAC2012-RULE\_20\_14-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_2" desc="The ', &quot;, /\* or // characters shall not occur in a header file name" id="MISRAC2012-RULE\_20\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_2" desc="The \ character should not occur in a header file name" id="MISRAC2012-RULE\_20\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_3" desc="The #include directive shall be followed by either a &lt;filename> or &quot;filename&quot; sequence" id="MISRAC2012-RULE\_20\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_4" desc="A macro shall not be defined with the same name as a keyword in C90" id="MISRAC2012-RULE\_20\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_4" desc="A macro shall not be defined with the same name as a keyword in C99" id="MISRAC2012-RULE\_20\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_5" desc="#undef shall not be used" id="MISRAC2012-RULE\_20\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_6" desc="Arguments to a function-like macro shall not contain tokens that look like preprocessing directives" id="MISRAC2012-RULE\_20\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_7" desc="In the definition of a function-like macro each instance of a parameter shall be enclosed in parentheses unless it is used as the operand of # or ##" id="MISRAC2012-RULE\_20\_7-a" sev="2">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_8" desc="The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1" id="MISRAC2012-RULE\_20\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_20\_9" desc="Do not use in preprocessor directives #if and #elif macros not defined in translation unit" id="MISRAC2012-RULE\_20\_9-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_1" desc="Do not #define or #undef identifiers with names which start with underscore" id="MISRAC2012-RULE\_21\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_1" desc="Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined (C90 code)" id="MISRAC2012-RULE\_21\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_1" desc="Reserved identifiers, macros and functions in the standard library, shall not be defined, redefined or undefined (C99 code)" id="MISRAC2012-RULE\_21\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_1" desc="Do not #define nor #undef identifier 'defined'" id="MISRAC2012-RULE\_21\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_10" desc="Avoid functions which use time from standard C library" id="MISRAC2012-RULE\_21\_10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_11" desc="The standard header file &lt;tgmath.h> shall not be used" id="MISRAC2012-RULE\_21\_11-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_11" desc="The facilities that are specified as being provided by &lt;tgmath.h> should not be used" id="MISRAC2012-RULE\_21\_11-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_12" desc="The exception handling features of &lt;fenv.h> should not be used" id="MISRAC2012-RULE\_21\_12-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_13" desc="Do not pass incorrect values to ctype.h library functions" id="MISRAC2012-RULE\_21\_13-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_14" desc="The Standard Library function memcmp shall not be used to compare null terminated strings" id="MISRAC2012-RULE\_21\_14-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_15" desc="The pointer arguments to the Standard Library functions 'memcmp', 'memmove' and 'memcmp' shall be pointers to qualified or unqualified versions of compatible types" id="MISRAC2012-RULE\_21\_15-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_16" desc="The pointer arguments to the Standard Library function 'memcmp' shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type" id="MISRAC2012-RULE\_21\_16-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_17" desc="Avoid overflow due to reading a not zero terminated string" id="MISRAC2012-RULE\_21\_17-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_17" desc="Avoid overflow when writing to a buffer" id="MISRAC2012-RULE\_21\_17-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_18" desc="The size\_t argument passed to any function in string.h shall have an appropriate value" id="MISRAC2012-RULE\_21\_18-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_19" desc="The pointers returned by the Standard Library functions 'localeconv', 'getenv', 'setlocale' or, 'strerror' shall only be used as if they have pointer to const-qualified type" id="MISRAC2012-RULE\_21\_19-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_19" desc="Strings pointed by members of the structure 'lconv' should not be modified" id="MISRAC2012-RULE\_21\_19-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_2" desc="The names of standard library macros, objects and functions shall not be reused" id="MISRAC2012-RULE\_21\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_2" desc="The names of standard library macros, objects and functions shall not be reused (C90)" id="MISRAC2012-RULE\_21\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_2" desc="The names of standard library macros, objects and functions shall not be reused (C99)" id="MISRAC2012-RULE\_21\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_20" desc="Pointers returned by certain Standard Library functions should not be used following a subsequent call to the same or related function" id="MISRAC2012-RULE\_21\_20-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_21" desc="The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="MISRAC2012-RULE\_21\_21-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_3" desc="Dynamic heap memory allocation shall not be used" id="MISRAC2012-RULE\_21\_3-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_4" desc="The setjmp macro and the longjmp function shall not be used" id="MISRAC2012-RULE\_21\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_4" desc="The standard header file &lt;setjmp.h> shall not be used" id="MISRAC2012-RULE\_21\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_5" desc="The standard header file &lt;signal.h> shall not be used" id="MISRAC2012-RULE\_21\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_5" desc="The signal handling facilities of &lt;signal.h> shall not be used" id="MISRAC2012-RULE\_21\_5-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_6" desc="The Standard Library input/output functions shall not be used" id="MISRAC2012-RULE\_21\_6-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_7" desc="The library functions atof, atoi and atol from library stdlib.h shall not be used" id="MISRAC2012-RULE\_21\_7-a" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_8" desc="The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="MISRAC2012-RULE\_21\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_8" desc="The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="MISRAC2012-RULE\_21\_8-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_8" desc="The 'quick\_exit()' and '\_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used" id="MISRAC2012-RULE\_21\_8-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_21\_9" desc="The library functions bsearch and qsort of &lt;stdlib.h> shall not be used" id="MISRAC2012-RULE\_21\_9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_1" desc="Ensure resources are freed" id="MISRAC2012-RULE\_22\_1-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_10" desc="Properly use errno value" id="MISRAC2012-RULE\_22\_10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_2" desc="Do not use resources that have been freed" id="MISRAC2012-RULE\_22\_2-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_2" desc="Do not free resources using invalid pointers" id="MISRAC2012-RULE\_22\_2-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_3" desc="The same file shall not be opened for read and write access at the same time on different streams" id="MISRAC2012-RULE\_22\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_4" desc="Avoid writing to a stream which has been opened as read only" id="MISRAC2012-RULE\_22\_4-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_5" desc="A pointer to a FILE object shall not be dereferenced" id="MISRAC2012-RULE\_22\_5-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_5" desc="A pointer to a FILE object shall not be dereferenced by a library function" id="MISRAC2012-RULE\_22\_5-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_6" desc="Do not use resources that have been freed" id="MISRAC2012-RULE\_22\_6-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_7" desc="The macro EOF should be compared with the unmodified return value from the Standard Library function" id="MISRAC2012-RULE\_22\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_8" desc="Properly use errno value" id="MISRAC2012-RULE\_22\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_22\_9" desc="Properly use errno value" id="MISRAC2012-RULE\_22\_9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_1" desc="There shall be no unreachable code in &quot;else&quot; block" id="MISRAC2012-RULE\_2\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_1" desc="There shall be no unreachable code after 'return', 'break', 'continue', and 'goto' statements" id="MISRAC2012-RULE\_2\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_1" desc="There shall be no unreachable code in &quot;if/else/while/for&quot; block" id="MISRAC2012-RULE\_2\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_1" desc="There shall be no unreachable code in switch statement" id="MISRAC2012-RULE\_2\_1-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_1" desc="There shall be no unreachable code in 'for' loop" id="MISRAC2012-RULE\_2\_1-e" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_1" desc="There shall be no unreachable code after 'if' or 'switch' statement" id="MISRAC2012-RULE\_2\_1-f" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_1" desc="There shall be no unreachable code after &quot;if&quot; or &quot;switch&quot; statement inside while/for/do...while loop" id="MISRAC2012-RULE\_2\_1-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_2" desc="All non-null statements shall either have at least one side-effect however executed or cause control flow to change" id="MISRAC2012-RULE\_2\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_2" desc="Avoid unused values" id="MISRAC2012-RULE\_2\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_3" desc="A function should not contain unused type declarations" id="MISRAC2012-RULE\_2\_3-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_3" desc="A source file should not contain unused type declarations" id="MISRAC2012-RULE\_2\_3-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_4" desc="A function should not contain unused local tag declarations" id="MISRAC2012-RULE\_2\_4-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_4" desc="A source file should not contain unused tag declarations" id="MISRAC2012-RULE\_2\_4-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_5" desc="A source file should not contain unused macro definitions" id="MISRAC2012-RULE\_2\_5-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_6" desc="A function should not contain unused label declarations" id="MISRAC2012-RULE\_2\_6-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_2\_7" desc="There should be no unused parameters in functions" id="MISRAC2012-RULE\_2\_7-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_3\_1" desc="The character sequence /\* shall not be used within a C-style comment" id="MISRAC2012-RULE\_3\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_3\_1" desc="The character sequence // shall not be used within a C-style comment" id="MISRAC2012-RULE\_3\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_3\_1" desc="The character sequence /\* shall not be used within a C++-style comment" id="MISRAC2012-RULE\_3\_1-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_3\_2" desc="Line-splicing shall not be used in // comments" id="MISRAC2012-RULE\_3\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_4\_1" desc="Octal and hexadecimal escape sequences shall be terminated" id="MISRAC2012-RULE\_4\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_4\_2" desc="Trigraphs shall not be used" id="MISRAC2012-RULE\_4\_2-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_1" desc="External identifiers shall be distinct" id="MISRAC2012-RULE\_5\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_2" desc="Identifiers declared in the file scope and in the same name space shall be distinct (c90)" id="MISRAC2012-RULE\_5\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_2" desc="Identifiers declared in the file scope and in the same name space shall be distinct (c99)" id="MISRAC2012-RULE\_5\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_2" desc="Identifiers declared in the same block scope and name space shall be distinct (c90)" id="MISRAC2012-RULE\_5\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_2" desc="Identifiers declared in the same block scope and name space shall be distinct (c99)" id="MISRAC2012-RULE\_5\_2-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_3" desc="Identifier declared in a local or function prototype scope shall not hide an identifier declared in a global or namespace scope" id="MISRAC2012-RULE\_5\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_3" desc="Identifiers declared in an inner local scope should not hide identifiers declared in an outer local scope" id="MISRAC2012-RULE\_5\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_4" desc="The name of a macro should be distinct from the names of its parameters(c90)" id="MISRAC2012-RULE\_5\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_4" desc="The name of a macro should be distinct from the names of its parameters(c99)" id="MISRAC2012-RULE\_5\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_4" desc="The name of a macro should be distinct from the names of other macros that are currently defined(c90)" id="MISRAC2012-RULE\_5\_4-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_4" desc="The name of a macro should be distinct from the names of other macros that are currently defined(c99)" id="MISRAC2012-RULE\_5\_4-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_5" desc="The names of macros that exist prior to preprocessing should be distinct from the identifiers that exist after preprocessing (c90)" id="MISRAC2012-RULE\_5\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_5" desc="The names of macros that exist prior to preprocessing should be distinct from the identifiers that exist after preprocessing (c99)" id="MISRAC2012-RULE\_5\_5-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_6" desc="Do not reuse typedef names" id="MISRAC2012-RULE\_5\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_6" desc="Do not reuse typedef names as a typedef name" id="MISRAC2012-RULE\_5\_6-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_7" desc="A tag name shall not be reused for other purpose within the program" id="MISRAC2012-RULE\_5\_7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_7" desc="A tag name shall not be reused to define a different tag" id="MISRAC2012-RULE\_5\_7-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_8" desc="Identifiers that define objects or functions with external linkage shall be unique" id="MISRAC2012-RULE\_5\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_9" desc="No object or function identifier with static storage duration should be reused" id="MISRAC2012-RULE\_5\_9-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_5\_9" desc="No object or function identifier with static storage duration should be reused" id="MISRAC2012-RULE\_5\_9-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_6\_1" desc="Bit fields shall only be defined to be of type unsigned int or signed int" id="MISRAC2012-RULE\_6\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_6\_2" desc="Named bit-fields with signed integer type shall have a length of more than one bit" id="MISRAC2012-RULE\_6\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_7\_1" desc="Octal constants (other than zero) shall not be used" id="MISRAC2012-RULE\_7\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_7\_2" desc="A 'U' suffix shall be applied to all constants of unsigned type" id="MISRAC2012-RULE\_7\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_7\_3" desc="Use capital 'L' instead of lowercase 'l' to indicate long" id="MISRAC2012-RULE\_7\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_7\_4" desc="A string literal shall not be modified" id="MISRAC2012-RULE\_7\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_1" desc="Whenever a function is declared or defined, its type shall be explicitly stated" id="MISRAC2012-RULE\_8\_1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_1" desc="Whenever an object is declared or defined, its type shall be explicitly stated" id="MISRAC2012-RULE\_8\_1-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_10" desc="An inline function shall be declared with the static storage class" id="MISRAC2012-RULE\_8\_10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_11" desc="When an array is declared with external linkage, its size shall be stated explicitly or defined implicitly by initialisation" id="MISRAC2012-RULE\_8\_11-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_12" desc="Within an enumerator list, the value of an implicitly-specified enumeration constant shall be unique" id="MISRAC2012-RULE\_8\_12-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_13" desc="A pointer parameter in a function prototype should be declared as pointer to const if the pointer is not used to modify the addressed object" id="MISRAC2012-RULE\_8\_13-a" sev="4">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_13" desc="Declare a type of parameter as typedef to pointer to const if the pointer is not used to modify the addressed object" id="MISRAC2012-RULE\_8\_13-b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_14" desc="The restrict type qualifier shall not be used" id="MISRAC2012-RULE\_8\_14-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_2" desc="Identifiers shall be given for all of the parameters in a function prototype declaration" id="MISRAC2012-RULE\_8\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_2" desc="Function types shall have named parameters" id="MISRAC2012-RULE\_8\_2-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_2" desc="Function types shall be in prototype form" id="MISRAC2012-RULE\_8\_2-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_3" desc="If objects or functions are declared more than once their types shall be compatible" id="MISRAC2012-RULE\_8\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_3" desc="The identifiers used in the declaration and definition of a function shall be identical" id="MISRAC2012-RULE\_8\_3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_3" desc="All declarations of an object or function shall have compatible types" id="MISRAC2012-RULE\_8\_3-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_4" desc="A declaration shall be visible when an object or function with external linkage is defined" id="MISRAC2012-RULE\_8\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_4" desc="If objects or functions are declared more than once their types shall be compatible" id="MISRAC2012-RULE\_8\_4-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_5" desc="An external object or function shall not have more than one non-defining declaration in translation unit" id="MISRAC2012-RULE\_8\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_6" desc="An identifier with external linkage shall have exactly one external definition" id="MISRAC2012-RULE\_8\_6-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_7" desc="Functions and objects should not be defined with external linkage if they are referenced in only one translation unit" id="MISRAC2012-RULE\_8\_7-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_8" desc="The static storage class specifier shall be used in definitions and declarations of objects and functions that have internal linkage" id="MISRAC2012-RULE\_8\_8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_8\_9" desc="Objects shall be defined at block scope if they are only accessed from within a single function" id="MISRAC2012-RULE\_8\_9-a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_9\_1" desc="Avoid use before initialization" id="MISRAC2012-RULE\_9\_1-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_9\_2" desc="The initializer for an aggregate or union shall be enclosed in braces" id="MISRAC2012-RULE\_9\_2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_9\_3" desc="Arrays shall not be partially initialized" id="MISRAC2012-RULE\_9\_3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_9\_4" desc="An element of an object shall not be initialized more than once" id="MISRAC2012-RULE\_9\_4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MISRAC2012-RULE\_9\_5" desc="Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly" id="MISRAC2012-RULE\_9\_5-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not allocate resources in function argument list because the order of evaluation of a function's parameters is undefined" id="MRM-01" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Use the same form in corresponding calls to new/malloc and delete/free" id="MRM-06" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not invoke malloc/realloc for objects having constructors" id="MRM-08" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Avoid hiding the global new" id="MRM-32" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Declare a copy assignment operator for classes with dynamically allocated memory" id="MRM-37" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Declare a copy constructor for classes with dynamically allocated memory" id="MRM-38" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Never return a dereferenced local pointer initialized by new in this function scope" id="MRM-23" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Call delete on pointer members in destructors" id="MRM-33" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Provide error handling for file opening errors right next to the call to fopen" id="MRM-39" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="A copy constructor shall copy all data members and bases" id="MRM-41" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Call fclose() on pointer member in destructor if the pointer was used to open a file" id="MRM-42" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="An assignment operator shall assign all data bases" id="MRM-43" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not pass a pointer that has insufficient storage capacity or that is not suitably aligned for the object being constructed to placement 'new'" id="MRM-55" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not allocate more than one resource in a single statement" id="MRM-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="All classes should contain the assignment operator or appropriate comment" id="MRM-04" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="MRM" desc="All classes should contain the copy constructor or appropriate comment" id="MRM-05" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="MRM" desc="Don't memcpy or memcmp non-PODs" id="MRM-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Always assign a new value to an expression that points to deallocated memory" id="MRM-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Always assign a new value to global or member variable that points to deallocated memory" id="MRM-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Always assign a new value to parameter or local variable that points to deallocated memory" id="MRM-11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Adhere to convention when writing new and delete" id="MRM-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Adhere to convention when writing new" id="MRM-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="If a class defines any overload of operator new, it should provide overloads of all three of plain, in-place, and non-throwing operator new" id="MRM-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="If a class defines any overload of operator new[], it should provide overloads of all three of plain, in-place, and non-throwing operator new[]" id="MRM-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="If a class defines any overload of operator delete, it should provide overloads of all three of plain, in-place, and non-throwing operator delete" id="MRM-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="If a class defines any overload of operator delete[], it should provide overloads of all three of plain, in-place, and non-throwing operator delete[]" id="MRM-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not allocate memory and expect that someone else will deallocate it later" id="MRM-18" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="MRM" desc="Do not allocate memory and expect that someone else will deallocate it later" id="MRM-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not allocate memory and expect that someone else will deallocate it later" id="MRM-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Use objects to manage resources" id="MRM-21" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Use objects to manage resources" id="MRM-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Store newed objects in smart pointers in standalone statements" id="MRM-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Write operator delete if you write operator new" id="MRM-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Write operator delete[] if you write operator new[]" id="MRM-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Always provide new and delete together" id="MRM-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Always provide new[] and delete[] together" id="MRM-29" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Use allocation by declaration rather than by new or malloc" id="MRM-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Freed memory shouldn't be accessed under any circumstances" id="MRM-31" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Destructor should not be called manually" id="MRM-31\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Check the return value of new" id="MRM-34" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="MRM" desc="Never provide brackets ([]) for delete when deallocating non-arrays" id="MRM-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Always provide empty brackets ([]) for delete when deallocating arrays" id="MRM-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Copy and destroy consistently" id="MRM-40" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="A copy assignment operator should be declared when a copy constructor is declared (and vice versa)" id="MRM-40\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Both the copy constructor and copy assignment operator should have the same public/protected/private permission" id="MRM-40\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="If you have a non-trivial copy constructor or copy assignment operator, you should also declare a destructor" id="MRM-40\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Both copy constructor and copy assignment operator should be declared for classes with a nontrivial destructor" id="MRM-40\_d" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not use sizeof operator on pointer type to specify the size of the memory to be allocated via 'malloc', 'calloc' or 'realloc' function" id="MRM-45" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not use calloc, malloc, realloc and free functions" id="MRM-46" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Classes containing at least one non-static member variable should declare the assignment operator or contain appropriate comment" id="MRM-47" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="MRM" desc="Classes containing at least one non-static member variable should declare the copy constructor or contain appropriate comment" id="MRM-48" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="MRM" desc="A copy constructor and a copy assignment operator shall be declared for classes that contain pointers to data items or nontrivial destructors" id="MRM-49" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Do not use 'delete' on pointers to a void type" id="MRM-51" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="The user defined 'new' operator should throw the 'std::bad\_alloc' exception when the allocation fails" id="MRM-53" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Avoid using the default operator 'new' for over-aligned types" id="MRM-54" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="An overhead should be used when an array of objects is passed to the placement 'new' allocation function" id="MRM-55\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Copy assignment operators should not have side effects that could affect copying the object" id="MRM-56" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Move assignment operators should not have side effects that could affect moving the object" id="MRM-57" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Declare both private copy constructor and copy assignment operator at the same time" id="MRM-50" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="MRM" desc="Use RAII to prevent resource leaks" id="MRM-52" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Identifiers for constant and enumerator values shall be lowercase" id="NAMING-42" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="All &quot;#define&quot; constants shall be in uppercase" id="NAMING-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="In an enumerated list, list members (elements) shall be in uppercase and names or tags for the list shall be in lowercase" id="NAMING-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Use lowercase for file names" id="NAMING-03" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="NAMING" desc="Global prefixes should only be used for global variables" id="NAMING-04" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin local variable names with a lowercase letters" id="NAMING-05" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin global variable names with a lowercase letters" id="NAMING-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin member variable names with a lowercase letters" id="NAMING-07" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="NAMING" desc="Begin all boolean type variables with 'b'" id="NAMING-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin class, struct, union, enum, and typedef names with an uppercase letter" id="NAMING-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="The names of abstract data types, structures, typedefs, and enumerated types are to begin with an uppercase letter" id="NAMING-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="The name of enumeration type shall begin with an uppercase letter and contain a suffix '\_t' at the end" id="NAMING-11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="The names of structures shall begin with an uppercase letter and contain a suffix '\_t' at the end" id="NAMING-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin constant variables with 'c'" id="NAMING-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin class data member names with 'its'" id="NAMING-14" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="NAMING" desc="Begin all double type variable with 'd'" id="NAMING-15" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="NAMING" desc="Begin all float type variables with 'f'" id="NAMING-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin all function names with uppercase letter" id="NAMING-17" sev="3">

<Stats authTot="14;" authUrg="0;" total="14"/>

</Rule>

<Rule cat="NAMING" desc="Begin global variable names with 'the'" id="NAMING-18" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="NAMING" desc="Begin all integer type variable with 'i'" id="NAMING-19" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="NAMING" desc="Functions that begin with 'is' should return boolean values" id="NAMING-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin all long integer variables with 'li'" id="NAMING-21" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Prefix a variable type 'pointer' with a 'p' character" id="NAMING-22" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING" desc="Begin all short integer variables with 'si'" id="NAMING-23" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin all signed character variables with 'c'" id="NAMING-24" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin all terminated characters string variables with 'sz'" id="NAMING-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin all unsigned character type variables with 'uc'" id="NAMING-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Begin all unsigned integer type variables with 'ui'" id="NAMING-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Use lowercase letters for structure and union member names" id="NAMING-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Append names of non-scalar typedefs with &quot;\_t&quot;" id="NAMING-29" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Implementation files in C always have the file name extension &quot;.c&quot;" id="NAMING-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Do not use typenames that differ only by the use of uppercase and lowercase letters" id="NAMING-31" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="An include file for a class should have a file name of the form &lt;class name> + extension" id="NAMING-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Do not use identifiers which begin with one or two underscores (`\_' or `\_\_')" id="NAMING-33" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="NAMING" desc="Global function names should start with lowercase" id="NAMING-34" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="NAMING" desc="Member function names should start with lowercase" id="NAMING-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Names of parameters in declaration and definition should be identical" id="NAMING-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Include files in C++ always have the file name extension '.hh'" id="NAMING-37" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING" desc="Implementation files in C++ always have the file name extension &quot;.cc&quot;" id="NAMING-38" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="NAMING" desc="Inline definition files always have the file name extension &quot;.icc&quot;" id="NAMING-39" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING" desc="Only the first word of the name of a class, structure, namespace, enumeration, or typedef will begin with an uppercase letter" id="NAMING-40" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING" desc="Header files will always have a file name extension of '.h'" id="NAMING-41" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING" desc="File name extension, if present, should be &quot;\*.cpp&quot; or &quot;\*.h&quot;" id="NAMING-43" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING" desc="All letters contained in function and variable names will be composed entirely of lowercase letters" id="NAMING-44" sev="3">

<Stats authTot="19;" authUrg="0;" total="19"/>

</Rule>

<Rule cat="NAMING" desc="Identifiers will not differ by mixture of case, the underscore character, interchange of the similarly looking letters and numbers" id="NAMING-45" sev="3">

<Stats authTot="20;" authUrg="0;" total="20"/>

</Rule>

<Rule cat="NAMING" desc="The ', &quot;, /\* or // characters shall not occur in a header file name" id="NAMING-46" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Different identifiers shall be typographically unambiguous" id="NAMING-47" sev="3">

<Stats authTot="13;" authUrg="0;" total="13"/>

</Rule>

<Rule cat="NAMING" desc="The \ character should not occur in a header file name" id="NAMING-48" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="User defined suffixes of the user defined literal operators shall start with underscore followed by one or more letters" id="NAMING-51" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Universal character names shall be used only inside character or string literals" id="NAMING-52" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Header files should have a file extension of: &quot;.h&quot;, &quot;.hpp&quot; or &quot;.hxx&quot;" id="NAMING-53" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Use visually distinct identifiers" id="NAMING-54" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Implementation files in C++ will always have a file name extension of &quot;.cpp&quot;" id="NAMING-49" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING" desc="Identifiers in the same name space with overlapping visibility should be typographically unambiguous" id="NAMING-50" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for array variables and parameters" id="NAMING-HN-01" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for bool types" id="NAMING-HN-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for bool pointer, array, or reference types" id="NAMING-HN-03" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for byte types" id="NAMING-HN-04" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for byte pointer, array, or reference types" id="NAMING-HN-05" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for char types" id="NAMING-HN-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for array of char types" id="NAMING-HN-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for pointer, array, or reference to array of char types" id="NAMING-HN-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for char pointer, array, or reference types" id="NAMING-HN-09" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for char pointer or reference types" id="NAMING-HN-10" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for constant parameters" id="NAMING-HN-11" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for double-precision floating point types" id="NAMING-HN-12" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for double-precision floating point pointer, array, or reference types" id="NAMING-HN-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for dword types" id="NAMING-HN-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for dword pointer, array, or reference types" id="NAMING-HN-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for dynamically allocated array" id="NAMING-HN-16" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for floating point types" id="NAMING-HN-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for floating point pointer, array, or reference types" id="NAMING-HN-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for class declaration" id="NAMING-HN-19" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for structs declaration" id="NAMING-HN-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for ifstream type variables and parameters" id="NAMING-HN-21" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for int types" id="NAMING-HN-22" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for int pointer, array, or reference types" id="NAMING-HN-23" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for istream type parameters and variables" id="NAMING-HN-24" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for long int types" id="NAMING-HN-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for long double-precision floating point types" id="NAMING-HN-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for long double-precision floating point pointer, array, or reference types" id="NAMING-HN-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for long int pointer, array, or reference types" id="NAMING-HN-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for member variables" id="NAMING-HN-29" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for int types" id="NAMING-HN-30" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for int pointer, array, or reference types" id="NAMING-HN-31" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for ofstream type parameters and variables" id="NAMING-HN-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for ostream type parameters and variables" id="NAMING-HN-33" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for pointer" id="NAMING-HN-34" sev="3">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for reference parameters" id="NAMING-HN-35" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for short int types" id="NAMING-HN-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for short int pointer, array, or reference types" id="NAMING-HN-37" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for static variables" id="NAMING-HN-38" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for string types" id="NAMING-HN-39" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for string pointer, array, or reference types" id="NAMING-HN-40" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for unsigned types" id="NAMING-HN-41" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for void pointer types" id="NAMING-HN-42" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for word types" id="NAMING-HN-43" sev="3">

<Stats authTot="6;" authUrg="0;" total="6"/>

</Rule>

<Rule cat="NAMING-HN" desc="Hungarian notation for word pointer, array, or reference types" id="NAMING-HN-44" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="OOP" desc="Class cannot inherit other class more than once unless it is virtual inheritance" id="OOP-03" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Be wary about using multiple inheritance of classes that are not abstract interfaces" id="OOP-07" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Do not directly access global data from a constructor" id="OOP-08" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid calling virtual functions from constructors" id="OOP-16" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid calling virtual functions from destructors" id="OOP-16\_b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Define a virtual destructor in classes used as base classes which have virtual functions" id="OOP-22" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="OOP" desc="Make destructors virtual in base classes" id="OOP-24" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A pointer to an abstract class shall not be converted to a pointer of a class that inherits from that abstract class" id="OOP-29" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Make base class destructors public and virtual, or protected and nonvirtual" id="OOP-31" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="If a class destructor is called and the class has virtual functions it shall have a virtual destructor" id="OOP-38" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Casts from a base class to a derived class should not be performed on polymorphic types" id="OOP-49" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Do not use multiple inheritance" id="OOP-05" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Multiple inheritance shall be limited to at most 1 protected implementation" id="OOP-07\_a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Multiple inheritance shall not use any public implementations" id="OOP-07\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid using global data in member functions" id="OOP-10" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid &quot;public&quot; data members" id="OOP-18" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid 'protected' data members" id="OOP-19" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="If a class has virtual functions it shall have a virtual destructor" id="OOP-23" sev="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="OOP" desc="Avoid casts down the inheritance hierarchy" id="OOP-28" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Down casting (casting from base to derived class) shall not be allowed" id="OOP-35" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A stateful virtual base shall be explicitly declared in each derived class that accesses it" id="OOP-39\_b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A base class shall not be both virtual and non-virtual in the same hierarchy" id="OOP-41" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid public copy constructors and assignment operators in base classes" id="OOP-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid slicing. Consider Clone instead of copying in base classes" id="OOP-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Do not derive functions with the same name from more than one base class" id="OOP-04" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="For multiple inheritance use virtual common base class" id="OOP-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid the use of global objects in destructors" id="OOP-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid using the friend mechanism" id="OOP-11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Friend declarations shall not be used except declarations of comparison operators" id="OOP-11\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Protected member function shall not return non-const handles to private class-data" id="OOP-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Do not redefine an inherited virtual function with a different default parameter value" id="OOP-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Avoid explicit cast from derived to a base class" id="OOP-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Use the virtual keyword if a subclass implements a virtual function" id="OOP-21" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="OOP" desc="Avoid declaring virtual functions inline" id="OOP-25" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="OOP" desc="Never convert pointers to objects of a derived class to pointers to objects of a virtual base class" id="OOP-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Declare copy assignment operator for class with reference or const members" id="OOP-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Declare the copy constructor and copy assignment operator private not in class itself, but in a specifically designed base class" id="OOP-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Never redefine an inherited nonvirtual function" id="OOP-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Do not redefine an inherited nonvirtual function with template parameter" id="OOP-33" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Public member functions shall not return non-const handles to private/protected class-data" id="OOP-36" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="OOP" desc="The copy assignment operator shall be declared protected or private in an abstract class" id="OOP-42" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="There shall be no more than one definition of each virtual function on each path through the inheritance hierarchy" id="OOP-44" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="All constructors that are callable with a single argument of fundamental type shall be declared explicit" id="OOP-45" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A copy constructor shall only initialize its base classes and the non-static members of the class of which it is a member" id="OOP-46" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Classes should not be derived from virtual bases" id="OOP-47" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Member data in non-POD types shall be private" id="OOP-48" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A pointer to a virtual base class shall only be cast to a pointer to a derived class by means of dynamic\_cast" id="OOP-50" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Member functions declared in derived class should not hide functions declared in base classes" id="OOP-53" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Do not increase the accessibility of overridden or hidden methods" id="OOP-54" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A user-defined assignment operator shall not be virtual" id="OOP-56" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Structs should only contain public data members and should not be a base or inherit" id="OOP-57" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Write a using declaration to redeclare overloaded functions" id="OOP-17" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Check for assignment to self in operator=" id="OOP-34" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A virtual base shall be explicitly declared in each derived class" id="OOP-39" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Hierarchies should be based on abstract classes" id="OOP-40" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A virtual function shall only be overridden by a pure virtual function if it is itself declared as pure virtual" id="OOP-43" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A pointer to a class may not be converted to a pointer of a second class unless the one inherits from the other" id="OOP-52" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="A non-POD type should be defined as class" id="OOP-55" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Consider use composition instead of private inheritance" id="OOP-14" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Prefer composition when don't need inheritance" id="OOP-37" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OOP" desc="Use namespace instead of class or structure containing only static functions" id="OOP-51" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Eliminate unused parameters" id="OPT-03" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="OPT" desc="Variables will not be introduced until they can be initialized with meaningful values" id="OPT-26" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Declare variables as locally as possible" id="OPT-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Prefer canonical form of ++ and --. Prefer calling the prefix forms" id="OPT-04" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="OPT" desc="Avoid unused private member variables" id="OPT-05" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="OPT" desc="Prefer &quot;a @= b&quot; than &quot;a = a @ b&quot;, where &quot;@&quot; is +, -, \*, /, %" id="OPT-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Prefer &quot;a @= b&quot; than &quot;a = a @ b&quot;, where &quot;@&quot; is &amp;, |, ^, &lt;&lt;, >>" id="OPT-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Do not declare variables in &quot;if&quot;, &quot;for&quot;, &quot;while&quot;, and &quot;do while&quot; statement" id="OPT-10" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="OPT" desc="If a file-level static variable is used/referenced in one function only then include that variable in the function itself" id="OPT-11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="If a file-level static variable is used/referenced in one class only then include that variable in the class itself" id="OPT-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Declare member variables in the descending size order" id="OPT-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Pass objects by reference instead of by value" id="OPT-14" sev="3">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Rule>

<Rule cat="OPT" desc="Consider overloading to avoid implicit type conversions" id="OPT-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Global function containing recursion, loops or virtual function call should not be inlined" id="OPT-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Avoid inline constructors and destructors" id="OPT-17" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="OPT" desc="Member function containing recursion or loops should not be inlined" id="OPT-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Consider using op= instead of stand-alone op" id="OPT-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Postpone variable definitions as long as possible" id="OPT-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Every switch statement shall have at least one non-empty case clause" id="OPT-21" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Useless case statement shall not be permitted" id="OPT-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="'strlen' function should not be used to check string against NULL/non-NULL" id="OPT-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Every defined function with internal linkage shall be used at least once" id="OPT-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="There shall be no unused parameters (named or unnamed) in non-virtual functions" id="OPT-31" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="OPT" desc="All non-empty functions with void return type shall have external side effect(s)" id="OPT-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Do not assign a variable to itself" id="OPT-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Do not use a variable inside its own initializer" id="OPT-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Every switch statement shall have at least two switch-clauses" id="OPT-39" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="There shall be no unused named parameters in virtual functions" id="OPT-42" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Avoid unused local variables" id="OPT-02" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Avoid unnecessary local variables" id="OPT-06" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Trivial accessor and mutator functions should be inlined" id="OPT-23" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Trivial forwarding functions should be inlined" id="OPT-24" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Only functions with 1 or 2 statements should be considered candidates for inline functions" id="OPT-25" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="The number of accessor and mutator functions should be minimized" id="OPT-27" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="OPT" desc="Redundant explicit cast to the same type is not allowed" id="OPT-29" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Functions with void return type shall not be empty" id="OPT-32\_b" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="OPT" desc="A function should not contain unused label declarations" id="OPT-37" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="There should be no unused parameters in functions" id="OPT-38" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="The same code (after preprocessing) in two branches of if-else-if chain" id="OPT-40\_a" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="The same code (after preprocessing) in different clauses of switch statement" id="OPT-40\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="A file should directly include only the headers that contain declarations and definitions required to compile that file" id="OPT-41" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Remove unnecessary '== true'" id="OPT-09" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OPT" desc="Consider returning object by reference instead of by value" id="OPT-33" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A1" desc="Protect against command injection" id="OWASP2017-A1-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A1" desc="Avoid printing tainted data on the output console" id="OWASP2017-A1-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A1" desc="Protect against environment injection" id="OWASP2017-A1-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A1" desc="Exclude unsanitized user input from format strings" id="OWASP2017-A1-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A1" desc="Protect against SQL injection" id="OWASP2017-A1-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A1" desc="Avoid passing unvalidated binary data to log methods" id="OWASP2017-A1-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A10" desc="All exceptions should be rethrown or logged with standard logger" id="OWASP2017-A10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A2" desc="Do not use weak encryption functions" id="OWASP2017-A2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A3" desc="Properly seed pseudorandom number generators" id="OWASP2017-A3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A4" desc="Disable resolving XML external entities (XXE) in libxerces-c" id="OWASP2017-A4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A5" desc="Protect against file name injection" id="OWASP2017-A5-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A5" desc="Observe correct revocation order while relinquishing privileges" id="OWASP2017-A5-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A5" desc="Ensure that privilege relinquishment is successful" id="OWASP2017-A5-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A6" desc="Properly use errno value" id="OWASP2017-A6-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A6" desc="Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class" id="OWASP2017-A6-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2017-A6" desc="Do not leave 'catch' blocks empty" id="OWASP2017-A6-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API10" desc="All exceptions should be rethrown or logged with standard logger" id="OWASP2019-API10-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API10" desc="Do not use 'syslog' function for logging purposes" id="OWASP2019-API10-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API2" desc="Do not use weak encryption functions" id="OWASP2019-API2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid buffer overflow due to defining incorrect format limits" id="OWASP2019-API3-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid overflow due to reading a not zero terminated string" id="OWASP2019-API3-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid overflow when reading from a buffer" id="OWASP2019-API3-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid overflow when writing to a buffer" id="OWASP2019-API3-g" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid buffer overflow from tainted data due to defining incorrect format limits" id="OWASP2019-API3-h" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid buffer read overflow from tainted data" id="OWASP2019-API3-i" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid buffer write overflow from tainted data" id="OWASP2019-API3-j" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid race conditions while checking for the existence of a symbolic link" id="OWASP2019-API3-m" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Do not pass empty container iterators to std algorithms as destinations" id="OWASP2019-API3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid accessing arrays out of bounds" id="OWASP2019-API3-b" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid accessing arrays and pointers out of bounds" id="OWASP2019-API3-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Properly seed pseudorandom number generators" id="OWASP2019-API3-k" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid passing sensitive data to functions that write to log files" id="OWASP2019-API3-l" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Do not print potentially sensitive information, resulting from an application error into exception messages" id="OWASP2019-API3-p" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Avoid functions which use time from MFC library" id="OWASP2019-API3-o" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="A pointer to a structure should not be passed to a function that can copy data to the user space" id="OWASP2019-API3-q" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API3" desc="Usage of system properties (environment variables) should be restricted" id="OWASP2019-API3-n" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API4" desc="Ensure resources are freed" id="OWASP2019-API4-b" sev="1">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Rule>

<Rule cat="OWASP2019-API4" desc="Validate potentially tainted data before it is used to determine the size of memory allocation" id="OWASP2019-API4-a" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="OWASP2019-API7" desc="Properly use errno value" id="OWASP2019-API7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API7" desc="Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class" id="OWASP2019-API7-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API7" desc="Do not leave 'catch' blocks empty" id="OWASP2019-API7-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Protect against command injection" id="OWASP2019-API8-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Avoid printing tainted data on the output console" id="OWASP2019-API8-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Protect against environment injection" id="OWASP2019-API8-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Protect against file name injection" id="OWASP2019-API8-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Exclude unsanitized user input from format strings" id="OWASP2019-API8-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Protect against SQL injection" id="OWASP2019-API8-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Disable resolving XML external entities (XXE) in libxerces-c" id="OWASP2019-API8-g" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Use care to ensure that LoadLibrary() will load the correct library" id="OWASP2019-API8-h" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Avoid passing dynamically created strings into exec" id="OWASP2019-API8-i" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API8" desc="Avoid passing user input into methods as parameters" id="OWASP2019-API8-j" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API9" desc="All usage of assembler shall be documented" id="OWASP2019-API9-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API9" desc="Use of floating-point arithmetic shall be documented" id="OWASP2019-API9-d" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="OWASP2019-API9" desc="All uses of the #pragma directive shall be documented and explained" id="OWASP2019-API9-g" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API9" desc="Objects or functions with external linkage shall be declared in a header file" id="OWASP2019-API9-e" sev="4">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="OWASP2019-API9" desc="Assert liberally to document internal assumptions and invariants" id="OWASP2019-API9-a" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API9" desc="When using enum, the values of each member should be explicitly declared" id="OWASP2019-API9-b" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2019-API9" desc="Document integer division" id="OWASP2019-API9-f" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A1" desc="Protect against file name injection" id="OWASP2021-A1-a" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A1" desc="Observe correct revocation order while relinquishing privileges" id="OWASP2021-A1-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A1" desc="Ensure that privilege relinquishment is successful" id="OWASP2021-A1-c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A2" desc="Properly seed pseudorandom number generators" id="OWASP2021-A2-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A3" desc="Protect against command injection" id="OWASP2021-A3-b" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A3" desc="Avoid printing tainted data on the output console" id="OWASP2021-A3-c" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A3" desc="Protect against environment injection" id="OWASP2021-A3-d" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A3" desc="Exclude unsanitized user input from format strings" id="OWASP2021-A3-e" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A3" desc="Protect against SQL injection" id="OWASP2021-A3-f" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A3" desc="Avoid passing unvalidated binary data to log methods" id="OWASP2021-A3-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A4" desc="Avoid passing sensitive data to functions that write to log files" id="OWASP2021-A4-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A5" desc="Properly use errno value" id="OWASP2021-A5-c" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A5" desc="Disable resolving XML external entities (XXE) in libxerces-c" id="OWASP2021-A5-d" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A5" desc="Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class" id="OWASP2021-A5-a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A5" desc="Do not leave 'catch' blocks empty" id="OWASP2021-A5-b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A7" desc="Do not use weak encryption functions" id="OWASP2021-A7-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A8" desc="Use care to ensure that LoadLibrary() will load the correct library" id="OWASP2021-A8-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="OWASP2021-A9" desc="All exceptions should be rethrown or logged with standard logger" id="OWASP2021-A9-a" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PARSER" desc="Parser warning" id="PARSER-WARNING" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PARSER" desc="Parser remark" id="PARSER-REMARK" sev="5">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="PB" desc="Do not call delete on non-pointers" id="PB-13" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Properly terminate character strings" id="PB-21" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The class object should be passed by reference if the class has non-static pointers and has no declared copy constructor" id="PB-23" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Operators should not return value by reference" id="PB-09" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The definition of a constructor shall not contain default arguments that produce a signature identical to that of the implicitly-declared copy constructor" id="PB-24" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Unsigned arithmetic shall not be used" id="PB-25" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Public and protected methods should not be invoked by class constructor" id="PB-26" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="A string literal shall not be modified" id="PB-27" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The following character sequences shall not appear in header file names: ', \, /\*, //, or &quot;" id="PB-28" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The left-hand operand of a right-shift operator shall not have a negative value" id="PB-29" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not cast from or to incomplete class at the point of casting" id="PB-54" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not delete objects with incomplete class at the point of deletion" id="PB-55" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Boolean condition always evaluates to the same value due to enumeration with only zero or only non-zero constants" id="PB-68" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not dereference pointer type expressions" id="PB-01" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="PB" desc="Do not use assignments inside a(b), a[b], and cast" id="PB-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Using mixed types in ternary operator is not allowed" id="PB-03" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Return value of a function must match declared return type" id="PB-05" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="PB" desc="Assignment operator should have operands of compatible types" id="PB-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not assign function return value to a variable of incompatible type" id="PB-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not assign signed constants to unsigned integer variables" id="PB-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Don't treat arrays polymorphically" id="PB-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Declared types of formal and actual parameters to functions must match" id="PB-11" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="PB" desc="Do not cast a signed char to an unsigned int" id="PB-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Incorrect End-Of-String (EOS) definition" id="PB-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Don't assign the dividend of two integers to a floating-point type" id="PB-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid unintentionally discarding the remainder of integer division" id="PB-15\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid assigning out-of-range value to char type" id="PB-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid assigning out-of-range value to unsigned char type" id="PB-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid overloading class methods on a pointer and a numerical type" id="PB-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not create inline non-member functions that contain local static data" id="PB-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid slicing function arguments / return value" id="PB-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not use increment and decrement expressions inside a(b), a[b], and cast" id="PB-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="More than one 'enum' type shall not be used as a switch condition or a label in a case statement" id="PB-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not call 'sizeof' on constants" id="PB-31" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not call 'sizeof' on a pointer type" id="PB-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid implicit type conversions in assignments involving enum types" id="PB-33" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid implicit type conversions in comparisons involving enum types" id="PB-33\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Third parameter to 'memcpy'/'strncpy'/'memmove' should not depend on second" id="PB-34" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Assignment operators shall not be used in conditions without brackets" id="PB-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The unbounded functions of library &lt;cstring> shall not be used" id="PB-37" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Narrow and wide string literals shall not be concatenated" id="PB-38" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="String literals with different encoding prefixes shall not be concatenated" id="PB-38\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="A function shall not return a reference or a pointer to a parameter that is passed by reference or const reference" id="PB-39" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="A function shall not return a pointer or a reference to a parameter that is passed by const reference" id="PB-39\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="An identifier with array type passed as a function argument shall not decay to a pointer" id="PB-41" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not pass expression with array type to a function with pointer or array type parameter" id="PB-41\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="An object shall not be assigned to an overlapping object" id="PB-42" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="All constructors of a class should explicitly call a constructor for all of its immediate base classes and all virtual base classes" id="PB-43" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The execution of a function registered with 'std::atexit()' or 'std::at\_quick\_exit()' should not exit via an exception" id="PB-44" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="There should be no mismatch between the '%s' and '%c' format specifiers in the format string and their corresponding arguments in the invocation of a string formatting function" id="PB-45" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="There should be no mismatch between the '%f' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="PB-46" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="There should be no mismatch between the '%i' and '%d' format specifiers in the string and their corresponding arguments in the invocation of a string formatting function" id="PB-47" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="There should be no mismatch between the '%u' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="PB-48" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="There should be no mismatch between the '%p' format specifier in the format string and its corresponding argument in the invocation of a string formatting function" id="PB-49" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The number of format specifiers in the format string and the number of corresponding arguments in the invocation of a string formatting function should be equal" id="PB-50" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Pointer arithmetic shall not be applied to pointers that address variables of non-array type" id="PB-51" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid overloading global functions on a pointer and a numerical type" id="PB-52" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid overloading namespace functions on a pointer and a numerical type" id="PB-53" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid implicit conversions from signed to unsigned type" id="PB-56" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="A pointer to an array of derived class objects should not be converted to a base class pointer" id="PB-57" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Within an enumerator list, the value of an implicitly-specified enumeration constant shall be unique" id="PB-58" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Missing comma in a string array initialization" id="PB-59" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Suspicious argument to malloc" id="PB-60" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Pointer arithmetic performed on freshly allocated memory" id="PB-61" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid function calls with incorrect argument order" id="PB-62" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Function address should not be compared to zero" id="PB-63" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The values returned by string comparison functions should be compared only to zero" id="PB-64" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid possible integer overflow in expressions in which the result is assigned to a variable of a wider integer type" id="PB-65" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid possible integer overflow in expressions in which the result is compared to an expression of a wider integer type" id="PB-65\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Integer overflow or underflow in constant expression in '+', '-', '\*' operator" id="PB-66\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The same code (after preprocessing) in both branches of if-else statement" id="PB-67" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="An element of an object shall not be initialized more than once" id="PB-69" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="An expression with enum underlying type shall only have values corresponding to the enumerators of the enumeration" id="PB-70" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not copy instances of structures containing a flexible array member" id="PB-71" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not call va\_arg with an argument of the incorrect type" id="PB-72" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="PB-75" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="PB-75\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="The 'quick\_exit()' and '\_Exit()' functions from the 'stdlib.h' or 'cstdlib' library shall not be used" id="PB-75\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="C-style strings shall not be used" id="PB-76" sev="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Rule>

<Rule cat="PB" desc="Expression statements shall not be explicit calls to constructors of temporary objects only" id="PB-77" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Use intmax\_t or uintmax\_t for formatted IO on programmer-defined integer types" id="PB-78" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Avoid function declarations that are syntactically ambiguous" id="PB-79" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Integer overflow or underflow in constant expression in '&lt;&lt;' operator" id="PB-66\_b" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Do not add or subtract a constant with a value greater than one from an iterator" id="PB-74" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PB" desc="Evaluation of constant unsigned integer expressions should not lead to wrap-around" id="PB-73" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PFO" desc="Don't define entities with linkage in a header file" id="PFO-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PFO" desc="Use multiple include guards" id="PFO-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PFO" desc="An include file should not contain more than one class definition" id="PFO-03" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="PFO" desc="File with extension '.c' may not include other files with extension '.c'" id="PFO-05" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PFO" desc="Never include other files in a file with extension '.icc'" id="PFO-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PFO" desc="Use multiple include guards with pattern based on the header file name" id="PFO-07" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="PFO" desc="Use mechanism that prevents multiple inclusion of the file i.e. include guards or &quot;#pragma once&quot; preprocessor directive" id="PFO-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PFO" desc="&quot;#pragma once&quot; preprocessor directive should not be used in source files" id="PFO-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PFO" desc="Do not define more than on class in a header file" id="PFO-10" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="PFO" desc="Every implementation file should declare a local constant string that describes the file" id="PFO-04" sev="5">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="PORT" desc="Don't use hard coded value of offset in structures" id="PORT-03" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Don't use hex constant '0xffffffff' as an error marker" id="PORT-04" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="A pointer to a long should not be casted to an int pointer" id="PORT-05" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not cast an int pointer to a long pointer" id="PORT-11" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Algorithms shall not make assumptions concerning the order of allocation of nonstatic data members separated by an access specifier" id="PORT-26" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Assigning result of ints operation to long without casting at least one of the ints is not allowed" id="PORT-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not assign a long int to an int" id="PORT-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not pass long casted to an int in function call" id="PORT-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Constant assignment to long should not involve int literals" id="PORT-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not initialize a long int with int literals" id="PORT-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not assign the value from 32 bit multiplication to long type" id="PORT-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="The type int shall always be declared as unsigned int or signed int" id="PORT-13" sev="3">

<Stats authTot="14;" authUrg="0;" total="14"/>

</Rule>

<Rule cat="PORT" desc="Avoid conversions from &quot;shorter&quot; type to &quot;longer&quot; type in assignment" id="PORT-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not cast from &quot;shorter&quot; type to &quot;longer&quot; type" id="PORT-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Avoid conversions from a &quot;shorter&quot; type to a &quot;longer&quot; type in function calls" id="PORT-16" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="PORT" desc="Don't use 'i64' or 'L' suffixes directly" id="PORT-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Instead of casting a long type operand of an expression to int cast the result of the expression" id="PORT-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not declare variables of long type directly" id="PORT-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not assign a long value to a double" id="PORT-21" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="The names of identifiers should not be longer than 64 characters" id="PORT-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Avoid implicit conversions to a type of narrower size" id="PORT-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Avoid conversions of constant values to a narrower type" id="PORT-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="A pointer to a structure should not be passed to a function that writes data to a file" id="PORT-29" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Type long double shall not be used" id="PORT-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Type wchar\_t shall not be used" id="PORT-33" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="User-specified C++ external identifiers should differ in the first 64 characters" id="PORT-34\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="User-specified C++ internal identifiers should differ from external identifiers in the first 64 characters" id="PORT-34\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="User-specified C++ internal identifiers declared in the same scope should differ in the first 64 characters" id="PORT-34\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="User-specified C external identifiers should differ in the first 31 characters" id="PORT-35\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="User-specified C internal identifiers should differ from external identifiers in the first 31 characters" id="PORT-35\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="User-specified C internal identifiers declared in the same scope should differ in the first 31 characters" id="PORT-35\_c" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="The names of identifiers should not be longer than 31 characters" id="PORT-36\_a" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="The names of macro identifiers should not be longer than 31 characters" id="PORT-36\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not throw an exception across execution boundaries" id="PORT-30" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Do not pass a nonstandard-layout type object across execution boundaries" id="PORT-31" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Use capital 'L' instead of lowercase 'l' to indicate long" id="PORT-01" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Use arrays of int types instead of large arrays of longs" id="PORT-22" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Use arrays of int types instead of dynamic allocated large arrays of longs" id="PORT-23" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PORT" desc="Don't use large arrays of pointers to bool, char, short, int and float types" id="PORT-24" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="Don't use macros in include statement" id="PREPROC-02" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="In a macro function, use parentheses before and after multiplication or division" id="PREPROC-05" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="Do not allow absolute or relative path names in #include statements" id="PREPROC-06" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="The #include directive shall use the &lt;filename.h> notation to include header files" id="PREPROC-09" sev="2">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="PREPROC" desc="Avoid using macro definitions" id="PREPROC-01" sev="3">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="PREPROC" desc="The #include pre-processor directive will only be used to include header (\*.h) files" id="PREPROC-08" sev="3">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Rule>

<Rule cat="PREPROC" desc="The #ifndef pre-processor directive will only be used to prevent multiple inclusions of the same header file" id="PREPROC-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="The #endif pre-processor directives will only be used to prevent multiple inclusions of the same header file" id="PREPROC-10\_b" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="PREPROC" desc="The #if pre-processor directive will only be used to prevent multiple inclusions of the same header file" id="PREPROC-10\_c" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="PREPROC" desc="The #ifdef, #else, #elif preprocessor directives should not be used" id="PREPROC-11" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="PREPROC" desc="The #if preprocessor directive should not be used" id="PREPROC-11\_b" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="PREPROC" desc="C++ macros shall only be used for include guards, type qualifiers, or storage class specifiers" id="PREPROC-12" sev="3">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Rule>

<Rule cat="PREPROC" desc="Macro should not contain token pasting, variable argument list nor recursive macro calls" id="PREPROC-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="All macros must expand into complete syntactic units" id="PREPROC-14" sev="3">

<Stats authTot="14;" authUrg="0;" total="14"/>

</Rule>

<Rule cat="PREPROC" desc="Pointer dereference operations may not be hidden in macro definitions" id="PREPROC-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="A macro parameter immediately following a # operator shall not immediately be followed by or preceded by a ## operator" id="PREPROC-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="The input/output library stdio.h shall not be included" id="PREPROC-18" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="PREPROC" desc="The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1" id="PREPROC-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="Match the filename in a #include directive to the one on the filesystem" id="PREPROC-20" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="#error directive shall not be used" id="PREPROC-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="The #pragma directive shall not be used" id="PREPROC-23" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="All macro identifiers in preprocessor directives shall be defined before use, except in #ifdef and #ifndef preprocessor directives and the defined() operator" id="PREPROC-24" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="#undef shall not be used" id="PREPROC-25" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="The #ifndef, #ifdef, #if, #elif, #else, and #endif pre-processor directives shall only be used for conditional file inclusion and include guards" id="PREPROC-26" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="PREPROC" desc="Don't redefine primitive types" id="PREPROC-03" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="Don't define part of statement" id="PREPROC-04" sev="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="PREPROC" desc="A macro parameter used as an operand to the # or ## operators, which is itself subject to further macro replacement, shall only be used as an operand to these operators" id="PREPROC-17" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="Incorrect 'NULL' definition" id="PREPROC-07" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="PREPROC" desc="Avoid token concatenation that may produce universal character names" id="PREPROC-21" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="The methods inside the SIGNAL and SLOT macros have to exist" id="QT-04" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Do not force thread termination by calling QThread::terminate()" id="QT-07" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Do not call new on QMutexLocker" id="QT-09" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Do not directly set specific Widget attributes that are set by Qt kernel" id="QT-12" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Every QObject subclass should contain a Q\_OBJECT macro" id="QT-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="QObject must not be inherited more than one time" id="QT-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="It's only possible to connect a signal to a slot or a signal to another signal" id="QT-03" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Mark signal emissions with the keyword &quot;emit&quot; to distinguish a signal emission from a normal method call" id="QT-05" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Do not set Idle or Critical priority to QThreads. Use High/Low/Med only" id="QT-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="If using lock on a mutex in a function, release the lock in the same function" id="QT-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Declare a copy constructor and operator= for any Qt - inherited objects that have pointers" id="QT-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Never mix const and non-const iterators in assignment" id="QT-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="If you use Q\_DECLARE\_FLAGS, you must also use Q\_DECLARE\_OPERATORS\_FOR\_FLAGS" id="QT-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Q\_DECLARE\_OPERATORS\_FOR\_FLAGS must be in the global namespace, Q\_DECLARE\_FLAGS inside the enum's namespace" id="QT-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="The getters and setters of a Q\_PROPERTY must exist and the types must match" id="QT-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Properties of enum types must use Q\_ENUMS" id="QT-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Non-const function should not be called on the Qt object" id="QT-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="QT" desc="Do not use setWindowFlags function on a Widget" id="QT-11" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Never use gets()" id="SECURITY-16" sev="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid functions which use time from standard C library" id="SECURITY-01" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid functions which use random numbers from standard C library" id="SECURITY-02" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Use care to ensure that LoadLibrary() will load the correct library" id="SECURITY-04" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid calling functions printf/wprintf with only one argument other than string constant" id="SECURITY-05" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using functions fprintf/fwprintf with only two parameters, when second parameter is a variable" id="SECURITY-08" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using data() function from 'string' class of standard library" id="SECURITY-09" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using the 'vfork()' function" id="SECURITY-10" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using unsecured shell functions that may be affected by shell metacharacters" id="SECURITY-11" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using unsafe string functions which may cause buffer overflows" id="SECURITY-12" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using unsafe string functions that do not check bounds" id="SECURITY-13" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use scanf and fscanf functions without specifying variable size in format string" id="SECURITY-14" sev="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="SECURITY" desc="Do not print potentially sensitive information, resulting from an application error into exception messages" id="SECURITY-15" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid passing non-const parameters or variables into exec" id="SECURITY-17" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid passing dynamically created strings into exec" id="SECURITY-18" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Usage of functions prone to race is not allowed" id="SECURITY-19" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid passing user input into methods as parameters" id="SECURITY-20" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use 'syslog' function for logging purposes" id="SECURITY-21" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use mbstowcs() function" id="SECURITY-22" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Beware of functions which may return the current directory or the windows directory" id="SECURITY-23" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using InitializeCriticalSection" id="SECURITY-24" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using thread-unsafe functions" id="SECURITY-25" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use 'setuid' in source code" id="SECURITY-26" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Don't use chmod(), chown(), chgrp()" id="SECURITY-27" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Standard random number generators should not be used to generate randomness for security reasons" id="SECURITY-28" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use obsolete C routine ulimit()" id="SECURITY-29" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using 'getpw' function in program code" id="SECURITY-30" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use 'cuserid' function" id="SECURITY-31" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using obsolete C routine 'usleep'" id="SECURITY-32" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Usage of functions which do not properly handle non-NULL terminated strings is not allowed" id="SECURITY-33" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid using environment variables" id="SECURITY-34" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Don't trust any value of command line if attacker can set them" id="SECURITY-35" sev="2">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Rule>

<Rule cat="SECURITY" desc="Never use unfiltered data from an untrusted user as the format parameter" id="SECURITY-36" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use weak encryption functions" id="SECURITY-37" sev="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Untrusted data is used as a loop boundary" id="SECURITY-38" sev="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use the rand() function for generating pseudorandom numbers" id="SECURITY-02\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Avoid functions which use time from MFC library" id="SECURITY-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Don't use unsafe C functions that do write to range-unchecked buffers" id="SECURITY-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Use secure temporary file name functions" id="SECURITY-39" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Specify the access permission bits if a file is created using the 'open' or 'openat' system call" id="SECURITY-42" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="The function 'pthread\_setcanceltype()' should not be called with 'PTHREAD\_CANCEL\_ASYNCHRONOUS' argument" id="SECURITY-43" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Observe correct revocation order while relinquishing privileges" id="SECURITY-44" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Ensure that privilege relinquishment is successful" id="SECURITY-45" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="A pointer to a structure should not be passed to a function that can copy data to the user space" id="SECURITY-46" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Use correct integer precisions when checking the right hand operand of the shift operator" id="SECURITY-47" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not call the 'system()' function from the 'stdlib.h' or 'cstdlib' library with an argument other than '0' (null pointer)" id="SECURITY-48" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="SECURITY-48\_b" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use the 'char' buffer to store input from 'std::cin'" id="SECURITY-51" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="The 'getenv()' function from the 'stdlib.h' or 'cstdlib' library shall not be used" id="SECURITY-52" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Usage of system properties (environment variables) should be restricted" id="SECURITY-03" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Call 'umask' before calling 'mkstemp'" id="SECURITY-40" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Call 'chdir' if you call 'chroot'" id="SECURITY-41" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Use the 'cnd\_signal()' function with a unique condition variable" id="SECURITY-49" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="SECURITY" desc="Do not use the 'notify\_one()' function when multiple threads are waiting on the same condition variable" id="SECURITY-50" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="To make copying efficient, correct, and immune to the slicing problem create containers of pointers instead of containers of objects" id="STL-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Call empty instead of checking size() against zero" id="STL-03" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Avoid using iterative calls to insert in an explicit loop" id="STL-04" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Almost all uses of copy where the destination range is specified using an insert iterator should be replaced with calls to range member functions" id="STL-05" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Instead of anonymous istream\_iterator objects use istream\_iterator names when used as function parameters" id="STL-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="When using containers of newed pointers, remember to delete the pointers before the container is destroyed" id="STL-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Never create containers of auto\_ptrs" id="STL-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Avoid using remove algorithm with list and standard associative containers" id="STL-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Prefer vector and string to dynamically allocated arrays" id="STL-10" sev="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Rule>

<Rule cat="STL" desc="Use reserve to avoid unnecessary reallocations" id="STL-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Each vector and string should be checked if it is not empty before it is passed to C function" id="STL-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Avoid using vector&lt;bool>" id="STL-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Specify comparison types for associative containers of pointers" id="STL-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="For associative containers never use comparison function returning true for equal values" id="STL-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="For associative containers never use comparison function returning true for equal values" id="STL-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Prefer iterator to const iterator, reverse\_iterator, and const\_reverse\_iterator" id="STL-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Use distance and advance to convert a container's const\_iterators to iterators" id="STL-19" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Make sure destination ranges are big enough" id="STL-22" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Follow remove-like algorithms by erase to remove elements from a container" id="STL-23" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Avoid using remove and similar algorithms (i.e., remove\_if and unique) on containers of dynamically allocated pointers" id="STL-24" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Proper implementation of copy\_if should not be based on returning the remove\_copy\_if with a not1 in front of the predicate" id="STL-26" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="If accumulate() is used on a container of floating point values, use floating point value as initial one" id="STL-27" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Design functor classes for pass-by-value" id="STL-28" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Make predicates const pure functions" id="STL-29" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Each functor class should has only one operator() function, and it's parameter and return types should be passed to unary\_function or binary\_function" id="STL-30" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="You must employ ptr\_fun, mem\_fun, or mem\_fun\_ref whenever you pass a function to an STL component" id="STL-31" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Make sure less&lt;T> means operator&lt;" id="STL-32" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Prefer algorithm calls to hand-written loops" id="STL-33" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Prefer member functions to algorithms with the same names" id="STL-34" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Do not rely on the conversion of count()'s nonzero values to true and zero to false" id="STL-35" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Do not use an iterator range that isn't really a range" id="STL-36" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="C-style arrays shall not be used" id="STL-37" sev="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Rule>

<Rule cat="STL" desc="Use != instead of &lt; to compare iterators" id="STL-38" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Use traits classes in conjunction with overloading" id="STL-39" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="When calling swap, employ a using declaration for std::swap, then call swap without namespace qualification" id="STL-40" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Do not declare the non-member to be an overloading of std::swap" id="STL-41" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Member version of swap should never throw exceptions" id="STL-42" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Consider using vector&lt;char> instead of string" id="STL-11" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="It is necessary to be careful when using reverse\_iterator's base iterator for erasure purposes" id="STL-20" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Consider istreambuf\_iterators for character-by-character input" id="STL-21" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="If you pass a sorted range to an algorithm that also takes a comparison function, be sure that the comparison function you pass behaves the same as the one you used to sort the range" id="STL-25" sev="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="STL" desc="Instead of trying to write the container-independent code use class encapsulation" id="STL-01" sev="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Factor parameter-independent code out of templates" id="TEMPL-01" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Define non-member functions inside templates when type conversions are desired" id="TEMPL-02" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Do not overload functions within a template class" id="TEMPL-03" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="In template global functions use 'typename' to identify nested dependent type names" id="TEMPL-04" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="In template classes use 'typename' to identify nested dependent type names" id="TEMPL-05" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Don't change default arguments of virtual functions in template classes" id="TEMPL-06" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="A copy constructor shall be declared when there is a template constructor with a single parameter that is a generic parameter" id="TEMPL-07" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="The viable function set for a function call should either contain no function specializations, or only contain function specializations" id="TEMPL-08" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Overloaded function templates shall not be explicitly specialized" id="TEMPL-09" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="All partial and explicit specializations for a template shall be declared in the same file as the declaration of their primary template" id="TEMPL-10" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="A copy assignment operator shall be declared when there is a template assignment operator with a parameter that is a generic parameter" id="TEMPL-11" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="In a class template with a dependent base, any name that may be found in that dependent base shall be referred to using a qualified-id or this->" id="TEMPL-12" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Do not declare non-member generic functions in associated namespaces" id="TEMPL-13" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="The function shall resolve to a function declared previously in the translation unit" id="TEMPL-14" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Declare 'extern' an explicitly instantiated template" id="TEMPL-15" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Template specialization shall be declared in the same file as the primary template or a user-defined type, for which the specialization is declared" id="TEMPL-16" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="Use a trailing return type syntax if the return type is preceded by the 'typename' keyword" id="TEMPL-17" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

<Rule cat="TEMPL" desc="A non-member generic operator shall only be declared in a namespace that does not contain class (struct) type, enum type or union type declarations" id="TEMPL-18" sev="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Rule>

</RulesList>

<CategoriesList>

<Category desc="DISA ASD STIG" name="APSC\_DV" total="104">

<Category desc="APSC\_DV-001850 The application must not display passwords/PINs as clear text." name="APSC\_DV-001850" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-001750 The application must transmit only cryptographically-protected passwords." name="APSC\_DV-001750" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-000480 The application must enforce approved authorizations for controlling the flow of information within the system based on organization-defined information flow control policies." name="APSC\_DV-000480" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-000160 The application must implement DoD-approved encryption to protect the confidentiality of remote access sessions." name="APSC\_DV-000160" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002460 The application must maintain the confidentiality and integrity of information during preparation for transmission." name="APSC\_DV-002460" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002560 The application must not be subject to input handling vulnerabilities." name="APSC\_DV-002560" total="11">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="APSC\_DV-003110 The application must not contain embedded authentication data." name="APSC\_DV-003110" total="1">

<Stats authTot="3;" authUrg="2;" total="3"/>

</Category>

<Category desc="APSC\_DV-002000 The application must terminate all network connections associated with a communications session at the end of the session." name="APSC\_DV-002000" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="APSC\_DV-002440 The application must protect the confidentiality and integrity of transmitted information." name="APSC\_DV-002440" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002540 The application must not be vulnerable to SQL Injection." name="APSC\_DV-002540" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-000500 The application must prevent non-privileged users from executing privileged functions to include disabling, circumventing, or altering implemented security safeguards/countermeasures." name="APSC\_DV-000500" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-001995 The application must not be vulnerable to race conditions." name="APSC\_DV-001995" total="9">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002520 The application must protect from canonical representation vulnerabilities." name="APSC\_DV-002520" total="11">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="APSC\_DV-003235 The application must not be subject to error handling vulnerabilities." name="APSC\_DV-003235" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002400 The application must restrict the ability to launch Denial of Service (DoS) attacks against itself or other information systems." name="APSC\_DV-002400" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002290 The application must use the Federal Information Processing Standard (FIPS) 140-2-validated cryptographic modules and random number generator if the application implements encryption, key exchange, digital signature, and hash functionality." name="APSC\_DV-002290" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002390 XML-based applications must mitigate DoS attacks by using XML filters, parser options, or gateways." name="APSC\_DV-002390" total="4">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="APSC\_DV-002590 The application must not be vulnerable to overflow attacks." name="APSC\_DV-002590" total="11">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-000170 The application must implement cryptographic mechanisms to protect the integrity of remote access sessions." name="APSC\_DV-000170" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-001300 The application must protect audit information from unauthorized deletion." name="APSC\_DV-001300" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-001740 The application must only store cryptographic representations of passwords." name="APSC\_DV-001740" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-000650 The application must not write sensitive data into the application logs." name="APSC\_DV-000650" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-001860 The application must use mechanisms meeting the requirements of applicable federal laws, Executive Orders, directives, policies, regulations, standards, and guidance for authentication to a cryptographic module." name="APSC\_DV-001860" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002470 The application must maintain the confidentiality and integrity of information during reception." name="APSC\_DV-002470" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002350 The application must use appropriate cryptography in order to protect stored DoD information when required by the information owner or DoD policy." name="APSC\_DV-002350" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002570 The application must generate error messages that provide information necessary for corrective actions without revealing information that could be exploited by adversaries." name="APSC\_DV-002570" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002010 The application must implement NSA-approved cryptography to protect classified information in accordance with applicable federal laws, Executive Orders, directives, policies, regulations, and standards." name="APSC\_DV-002010" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002550 The application must not be vulnerable to XML-oriented attacks." name="APSC\_DV-002550" total="11">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Category>

<Category desc="APSC\_DV-002530 The application must validate all input." name="APSC\_DV-002530" total="11">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="APSC\_DV-002510 The application must protect from command injection." name="APSC\_DV-002510" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-002480 The application must not disclose unnecessary information to users." name="APSC\_DV-002480" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="APSC\_DV-001290 The application must protect audit information from unauthorized modification." name="APSC\_DV-001290" total="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR C++14 Coding Guidelines" name="AUTOSAR" total="554">

<Category desc="AUTOSAR A16-6-1 #error directive shall not be used" name="AUTOSAR-A16\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-3-1 All overloads of a function shall be visible from where it is called" name="AUTOSAR-A7\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M8-5-2 Braces shall be used to indicate and match the structure in the non-zero initialization of arrays and structures" name="AUTOSAR-M8\_5\_2" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A6-5-1 A for-loop that loops through all elements of the container and does not use its loop-counter shall not be used" name="AUTOSAR-A6\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A25-4-1 Ordering predicates used with associative containers and STL sorting and related algorithms shall adhere to a strict weak ordering relation" name="AUTOSAR-A25\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-1-1 The assignment-expression of a throw statement shall not itself cause an exception to be thrown" name="AUTOSAR-M15\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-1-3 An empty throw (throw;) shall only be used in the compound statement of a catch handler" name="AUTOSAR-M15\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-1-2 NULL shall not be thrown explicitly" name="AUTOSAR-M15\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-7-1 If the behavior of a user-defined special member function is identical to implicitly defined special member function, then it shall be defined &quot;=default&quot; or be left undefined" name="AUTOSAR-A12\_7\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-3-2 The unary minus operator shall not be applied to an expression whose underlying type is unsigned" name="AUTOSAR-M5\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-3-1 Each operand of the ! operator, the logical &amp;&amp; or the logical || operators shall have type bool" name="AUTOSAR-M5\_3\_1" total="1">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="AUTOSAR M5-3-4 Evaluation of the operand to the sizeof operator shall not contain side effects" name="AUTOSAR-M5\_3\_4" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-3-3 The unary &amp; operator shall not be overloaded" name="AUTOSAR-M5\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-5-2 All user-defined conversion operators shall be defined explicit" name="AUTOSAR-A13\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-5-3 User-defined conversion operators should not be used" name="AUTOSAR-A13\_5\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A6-6-1 The goto statement shall not be used" name="AUTOSAR-A6\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M27-0-1 The stream input/output library &lt;cstdio> shall not be used" name="AUTOSAR-M27\_0\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A13-5-1 If &quot;operator[]&quot; is to be overloaded with a non-const version, const version shall also be implemented" name="AUTOSAR-A13\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-16-1 The ternary conditional operator shall not be used as a sub-expression" name="AUTOSAR-A5\_16\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-5-4 If two opposite operators are defined, one shall be defined in terms of the other" name="AUTOSAR-A13\_5\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-5-5 Comparison operators shall be non-member functions with identical parameter types and noexcept" name="AUTOSAR-A13\_5\_5" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A23-0-2 Elements of a container shall only be accessed via valid references, iterators, and pointers" name="AUTOSAR-A23\_0\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-1-4 If a function exits with an exception, then before a throw, the function shall place all objects/resources that the function constructed in valid states or it shall delete them" name="AUTOSAR-A15\_1\_4" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A23-0-1 An iterator shall not be implicitly converted to const\_iterator" name="AUTOSAR-A23\_0\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M4-5-3 Expressions with type (plain) char and wchar\_t shall not be used as operands to built-in operators other than the assignment operator =, the equality operators == and !=, and the unary &amp; operator" name="AUTOSAR-M4\_5\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-1-5 Exceptions shall not be thrown across execution boundaries" name="AUTOSAR-A15\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-17-1 The semantic equivalence between a binary operator and its assignment operator form shall be preserved" name="AUTOSAR-M5\_17\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-1-2 An exception object shall not be a pointer" name="AUTOSAR-A15\_1\_2" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR M4-5-1 Expressions with type bool shall not be used as operands to built-in operators other than the assignment operator =, the logical operators &amp;&amp;, ||, !, the equality operators == and !=, the unary &amp; operator, and the conditional operator" name="AUTOSAR-M4\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-1-10 Every defined function should be called at least once" name="AUTOSAR-M0\_1\_10" total="2">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Category>

<Category desc="AUTOSAR A15-1-1 Only instances of types derived from std::exception shall be thrown" name="AUTOSAR-A15\_1\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A10-4-1 Hierarchies should be based on interface classes" name="AUTOSAR-A10\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A16-7-1 The #pragma directive shall not be used" name="AUTOSAR-A16\_7\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A11-0-1 A non-POD type should be defined as class" name="AUTOSAR-A11\_0\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A11-0-2 A type defined as struct shall: (1) provide only public data members, (2) not provide any special member functions or methods, (3) not be a base of another struct or class, (4) not inherit from another struct or class" name="AUTOSAR-A11\_0\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-2-3 Enumerations shall be declared as scoped enum classes" name="AUTOSAR-A7\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M8-4-4 A function identifier shall either be used to call the function or it shall be preceded by &amp;" name="AUTOSAR-M8\_4\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-2-2 Enumeration underlying base type shall be explicitly defined" name="AUTOSAR-A7\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-2-1 An expression with enum underlying type shall only have values corresponding to the enumerators of the enumeration" name="AUTOSAR-A7\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M8-4-2 The identifiers used for the parameters in a re-declaration of a function shall be identical to those in the declaration" name="AUTOSAR-M8\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-7-1 The character \ shall not occur as a last character of a C++ comment" name="AUTOSAR-A2\_7\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M3-9-1 The types used for an object, a function return type, or a function parameter shall be token-for-token identical in all declarations and re-declarations" name="AUTOSAR-M3\_9\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-7-3 All declarations of &quot;user-defined&quot; types, static and non-static data members, functions and methods shall be preceded by documentation using &quot;///&quot; comments and &quot;@tag&quot; tags" name="AUTOSAR-A2\_7\_3" total="2">

<Stats authTot="65;" authUrg="0;" total="65"/>

</Category>

<Category desc="AUTOSAR A2-7-2 Sections of code shall not be &quot;commented out&quot;" name="AUTOSAR-A2\_7\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M12-1-1 An object's dynamic type shall not be used from the body of its constructor or destructor" name="AUTOSAR-M12\_1\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M3-9-3 The underlying bit representations of floating-point values shall not be used" name="AUTOSAR-M3\_9\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A1-1-1 All code shall conform to ISO/IEC 14882:2014 - Programming Language C++ and shall not use deprecated features" name="AUTOSAR-A1\_1\_1" total="11">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR M3-1-2 Functions shall not be declared at block scope" name="AUTOSAR-M3\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A6-4-1 A switch statement shall have at least two case-clauses, distinct from the default label" name="AUTOSAR-A6\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-6-1 All class data members that are initialized by the constructor shall be initialized using member initializers" name="AUTOSAR-A12\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-0-3 Control shall not be transferred into a try or catch block using a goto or a switch statement" name="AUTOSAR-M15\_0\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-2-6 A cast shall not convert a pointer to a function to any other pointer type, including a pointer to function type" name="AUTOSAR-M5\_2\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-2-9 A cast shall not convert a pointer type to an integral type" name="AUTOSAR-M5\_2\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M14-5-3 A copy assignment operator shall be declared when there is a template assignment operator with a parameter that is a generic parameter" name="AUTOSAR-M14\_5\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M18-2-1 The macro offsetof shall not be used" name="AUTOSAR-M18\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-2-8 An object with integer type or pointer to void type shall not be converted to an object with pointer type" name="AUTOSAR-M5\_2\_8" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A5-6-1 The right hand operand of the integer division or remainder operators shall not be equal to zero" name="AUTOSAR-A5\_6\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR M5-2-3 Casts from a base class to a derived class should not be performed on polymorphic types" name="AUTOSAR-M5\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-2-2 A pointer to a virtual base class shall only be cast to a pointer to a derived class by means of dynamic\_cast" name="AUTOSAR-M5\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A6-5-3 Do statements should not be used" name="AUTOSAR-A6\_5\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A6-5-2 A for loop shall contain a single loop-counter which shall not have floating-point type" name="AUTOSAR-A6\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-13-5 Hexadecimal constants should be upper case" name="AUTOSAR-A2\_13\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-13-6 Universal character names shall be used only inside character or string literals" name="AUTOSAR-A2\_13\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A6-5-4 For-init-statement and expression should not perform actions other than loop-counter initialization and modification" name="AUTOSAR-A6\_5\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-13-3 Type wchar\_t shall not be used" name="AUTOSAR-A2\_13\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-13-4 String literals shall not be assigned to non-constant pointers" name="AUTOSAR-A2\_13\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-13-1 Only those escape sequences that are defined in ISO/IEC 14882:2014 shall be used" name="AUTOSAR-A2\_13\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-13-2 String literals with different encoding prefixes shall not be concatenated" name="AUTOSAR-A2\_13\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-15 Array indexing shall be the only form of pointer arithmetic" name="AUTOSAR-M5\_0\_15" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR M5-0-16 A pointer operand and any pointer resulting from pointer arithmetic using that operand shall both address elements of the same array" name="AUTOSAR-M5\_0\_16" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-17 Subtraction between pointers shall only be applied to pointers that address elements of the same array" name="AUTOSAR-M5\_0\_17" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-18 >, >=, &lt;, &lt;= shall not be applied to objects of pointer type, except where they point to the same array" name="AUTOSAR-M5\_0\_18" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-0-2 At least the basic guarantee for exception safety shall be provided for all operations. In addition, each function may offer either the strong guarantee or the nothrow guarantee" name="AUTOSAR-A15\_0\_2" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR M9-3-1 Const member functions shall not return non-const pointers or references to class-data" name="AUTOSAR-M9\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-10 If the bitwise operators ~ and &lt;&lt; are applied to an operand with an underlying type of unsigned char or unsigned short, the result shall be immediately cast to the underlying type of the operand" name="AUTOSAR-M5\_0\_10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M9-3-3 If a member function can be made static then it shall be made static, otherwise if it can be made const then it shall be made const" name="AUTOSAR-M9\_3\_3" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR M5-0-11 The plain char type shall only be used for the storage and use of character values" name="AUTOSAR-M5\_0\_11" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-12 Signed char and unsigned char type shall only be used for the storage and use of numeric values" name="AUTOSAR-M5\_0\_12" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-14 The first operand of a conditional-operator shall have type bool" name="AUTOSAR-M5\_0\_14" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-1-2 A pointer or reference parameter in a function shall be declared as pointer to const or reference to const if the corresponding object is not modified" name="AUTOSAR-M7\_1\_2" total="3">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="AUTOSAR A7-1-4 The register keyword shall not be used" name="AUTOSAR-A7\_1\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-1-3 CV-qualifiers shall be placed on the right hand side of the type that is a typedef or a using name" name="AUTOSAR-A7\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-1-2 The constexpr specifier shall be used for values that can be determined at compile time" name="AUTOSAR-A7\_1\_2" total="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A7-1-1 Constexpr or const specifiers shall be used for immutable data declaration" name="AUTOSAR-A7\_1\_1" total="1">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Category>

<Category desc="AUTOSAR A2-8-1 A header file name shall be identical to a type name declared in it if it declares a type" name="AUTOSAR-A2\_8\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-11-1 Volatile keyword shall not be used" name="AUTOSAR-A2\_11\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A9-5-1 Unions shall not be used" name="AUTOSAR-A9\_5\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-3-1 The statement forming the body of a switch, while, do ... while or for statement shall be a compound statement" name="AUTOSAR-M6\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-20 Non-constant operands to a binary bitwise operator shall have the same underlying type" name="AUTOSAR-M5\_0\_20" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-21 Bitwise operators shall only be applied to operands of unsigned underlying type" name="AUTOSAR-M5\_0\_21" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-0-1 The C library facilities shall only be accessed through C++ library headers" name="AUTOSAR-A18\_0\_1" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A18-0-3 The library &lt;clocale> (locale.h) and the setlocale function shall not be used" name="AUTOSAR-A18\_0\_3" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-4-2 Use of floating-point arithmetic shall be documented" name="AUTOSAR-M0\_4\_2" total="1">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="AUTOSAR A18-0-2 The error state of a conversion from string to a numeric value shall be checked" name="AUTOSAR-A18\_0\_2" total="2">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Category>

<Category desc="AUTOSAR A13-3-1 A function that contains &quot;forwarding reference&quot; as its argument shall not be overloaded" name="AUTOSAR-A13\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-5-1 A pointer to member shall not access non-existent class members" name="AUTOSAR-A5\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M14-6-1 In a class template with a dependent base, any name that may be found in that dependent base shall be referred to using a qualified-id or this->" name="AUTOSAR-M14\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A0-4-2 Type long double shall not be used" name="AUTOSAR-A0\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A0-4-4 Range, domain and pole errors shall be checked when using math functions" name="AUTOSAR-A0\_4\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A4-7-1 An integer expression shall not lead to data loss" name="AUTOSAR-A4\_7\_1" total="8">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-2-5 Enumerations should be used to represent sets of related named constants" name="AUTOSAR-A7\_2\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-2-4 In an enumeration, either (1) none, (2) the first or (3) all enumerators shall be initialized" name="AUTOSAR-A7\_2\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A20-8-6 std::make\_shared shall be used to construct objects owned by std::shared\_ptr" name="AUTOSAR-A20\_8\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A20-8-5 std::make\_unique shall be used to construct objects owned by std::unique\_ptr" name="AUTOSAR-A20\_8\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A20-8-7 A std::weak\_ptr shall be used to represent temporary shared ownership" name="AUTOSAR-A20\_8\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A20-8-2 A std::unique\_ptr shall be used to represent exclusive ownership" name="AUTOSAR-A20\_8\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A20-8-1 An already-owned pointer value shall not be stored in an unrelated smart pointer" name="AUTOSAR-A20\_8\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A20-8-4 A std::unique\_ptr shall be used over std::shared\_ptr if ownership sharing is not required" name="AUTOSAR-A20\_8\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A20-8-3 A std::shared\_ptr shall be used to represent shared ownership" name="AUTOSAR-A20\_8\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A9-6-1 Data types used for interfacing with hardware or conforming to communication protocols shall be trivial, standard-layout and only contain members of types with defined sizes" name="AUTOSAR-A9\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-2-3 Before preprocessing, a null statement shall only occur on a line by itself; it may be followed by a comment, provided that the first character following the null statement is a white-space character" name="AUTOSAR-M6\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-2-2 Floating-point expressions shall not be directly or indirectly tested for equality or inequality" name="AUTOSAR-M6\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-4-1 Destructor of a base class shall be public virtual, public override or protected non-virtual" name="AUTOSAR-A12\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-10-1 An identifier declared in an inner scope shall not hide an identifier declared in an outer scope" name="AUTOSAR-A2\_10\_1" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-2-1 Assignment operators shall not be used in sub-expressions" name="AUTOSAR-M6\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A6-2-2 Expression statements shall not be explicit calls to constructors of temporary objects only" name="AUTOSAR-A6\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A6-2-1 Move and copy assignment operators shall either move or respectively copy base classes and data members of a class, without any side effects" name="AUTOSAR-A6\_2\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A9-6-2 Bit-fields shall be used only when interfacing to hardware or conforming to communication protocols" name="AUTOSAR-A9\_6\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-4-2 If a public destructor of a class is non-virtual, then the class should be declared final" name="AUTOSAR-A12\_4\_2" total="1">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Category>

<Category desc="AUTOSAR A18-5-11 &quot;operator new&quot; and &quot;operator delete&quot; shall be defined together" name="AUTOSAR-A18\_5\_11" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-11 A smart pointer shall only be used as a parameter type if it expresses lifetime semantics" name="AUTOSAR-A8\_4\_11" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-5-10 Placement new shall be used only with properly aligned pointers to sufficient storage capacity" name="AUTOSAR-A18\_5\_10" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-10 A parameter shall be passed by reference if it can't be NULL" name="AUTOSAR-A8\_4\_10" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A8-4-13 A std::shared\_ptr shall be passed to a function as: (1) a copy to express the function shares ownership (2) an lvalue reference to express that the function replaces the managed object (3) a const lvalue reference to express that the function retains a reference count" name="AUTOSAR-A8\_4\_13" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-12 A std::unique\_ptr shall be passed to a function as: (1) a copy to express the function assumes ownership (2) an lvalue reference to express that the function replaces the managed object" name="AUTOSAR-A8\_4\_12" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-1-2 The std::vector&lt;bool> specialization shall not be used" name="AUTOSAR-A18\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-1-1 C-style arrays shall not be used" name="AUTOSAR-A18\_1\_1" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A13-2-1 An assignment operator shall return a reference to &quot;this&quot;" name="AUTOSAR-A13\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-1-4 A pointer pointing to an element of an array of objects shall not be passed to a smart pointer of single object type" name="AUTOSAR-A18\_1\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M18-0-3 The library functions abort, exit, getenv and system from library &lt;cstdlib> shall not be used" name="AUTOSAR-M18\_0\_3" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-2-2 A binary arithmetic operator and a bitwise operator shall return a &quot;prvalue&quot;" name="AUTOSAR-A13\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-1-3 The std::auto\_ptr type shall not be used" name="AUTOSAR-A18\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-1-6 All std::hash specializations for user-defined types shall have a noexcept function call operator" name="AUTOSAR-A18\_1\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-2-3 A relational operator shall return a boolean value" name="AUTOSAR-A13\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M18-0-5 The unbounded functions of library &lt;cstring> shall not be used" name="AUTOSAR-M18\_0\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M18-0-4 The time handling functions of library &lt;ctime> shall not be used" name="AUTOSAR-M18\_0\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M2-10-1 Different identifiers shall be typographically unambiguous" name="AUTOSAR-M2\_10\_1" total="1">

<Stats authTot="13;" authUrg="0;" total="13"/>

</Category>

<Category desc="AUTOSAR M5-14-1 The right hand operand of a logical &amp;&amp;, || operators shall not contain side effects" name="AUTOSAR-M5\_14\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-1-9 A class, structure, or enumeration shall not be declared in the definition of its type" name="AUTOSAR-A7\_1\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-1-8 A non-type specifier shall be placed before a type specifier in a declaration" name="AUTOSAR-A7\_1\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-1-7 Each expression statement and identifier declaration shall be placed on a separate line" name="AUTOSAR-A7\_1\_7" total="3">

<Stats authTot="14;" authUrg="0;" total="14"/>

</Category>

<Category desc="AUTOSAR A7-1-6 The typedef specifier shall not be used" name="AUTOSAR-A7\_1\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-1-5 The auto specifier shall not be used apart from following cases: (1) to declare that a variable has the same type as return type of a function call, (2) to declare that a variable has the same type as initializer of non-fundamental type, (3) to declare parameters of a generic lambda expression, (4) to declare a function template using trailing return type syntax" name="AUTOSAR-A7\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-9-2 Forwarding values to other functions shall be done via: (1) std::move if the value is an rvalue reference, (2) std::forward if the value is forwarding reference" name="AUTOSAR-A18\_9\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-9-1 The std::bind shall not be used" name="AUTOSAR-A18\_9\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-9-4 An argument to std::forward shall not be subsequently used" name="AUTOSAR-A18\_9\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-9-3 The std::move shall not be used on objects declared const or const&amp;" name="AUTOSAR-A18\_9\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A14-8-2 Explicit specializations of function templates shall not be used" name="AUTOSAR-A14\_8\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-8-1 An object shall not be accessed outside of its lifetime" name="AUTOSAR-A3\_8\_1" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-3-2 The # and ## operators should not be used" name="AUTOSAR-M16\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-5-4 If a class has a user-declared constructor that takes a parameter of type std::initializer\_list, then it shall be the only constructor apart from special member function constructors" name="AUTOSAR-A8\_5\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-3-1 There shall be at most one occurrence of the # or ## operators in a single macro definition" name="AUTOSAR-M16\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-3-2 The identifier main shall not be used for a function other than the global function main" name="AUTOSAR-M7\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-3-1 The global namespace shall only contain main, namespace declarations and extern &quot;C&quot; declarations" name="AUTOSAR-M7\_3\_1" total="1">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Category>

<Category desc="AUTOSAR A16-2-1 The ', &quot;, /\*, //, \ characters shall not occur in a header file name or in #include directive" name="AUTOSAR-A16\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-5-1 In an initialization list, the order of initialization shall be following: (1) virtual base classes in depth and left to right order of the inheritance graph, (2) direct base classes in left to right order of inheritance list, (3) non-static data members in the order they were declared in the class definition" name="AUTOSAR-A8\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-3-4 Using-directives shall not be used" name="AUTOSAR-M7\_3\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-5-0 All memory shall be initialized before it is read" name="AUTOSAR-A8\_5\_0" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-3-3 There shall be no unnamed namespaces in header files" name="AUTOSAR-M7\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-5-3 A variable of type auto shall not be initialized using {} or ={} braced-initialization" name="AUTOSAR-A8\_5\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-3-6 Using-directives and using-declarations (excluding class scope or function scope using-declarations) shall not be used in header files" name="AUTOSAR-M7\_3\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A16-2-2 There shall be no unused include directives" name="AUTOSAR-A16\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-5-2 Braced-initialization {}, without equals sign, shall be used for variable initialization" name="AUTOSAR-A8\_5\_2" total="1">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Category>

<Category desc="AUTOSAR A17-0-1 Reserved identifiers, macros and functions in the C++ standard library shall not be defined, redefined or undefined" name="AUTOSAR-A17\_0\_1" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A9-3-1 Member functions shall not return non-const &quot;raw&quot; pointers or references to private or protected data owned by the class" name="AUTOSAR-A9\_3\_1" total="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR M6-5-5 A loop-control-variable other than the loop-counter shall not be modified within condition or expression" name="AUTOSAR-M6\_5\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-5-4 The loop-counter shall be modified by one of: --, ++, -= n, or += n; where n remains constant for the duration of the loop" name="AUTOSAR-M6\_5\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M3-4-1 An identifier declared to be an object or type shall be defined in a block that minimizes its visibility" name="AUTOSAR-M3\_4\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-5-3 The loop-counter shall not be modified within condition or statement" name="AUTOSAR-M6\_5\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-5-2 If loop-counter is not modified by -- or ++, then, within condition, the loop-counter shall only be used as an operand to &lt;=, &lt;, > or >=" name="AUTOSAR-M6\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-5-6 A loop-control-variable other than the loop-counter which is modified in statement shall have type bool" name="AUTOSAR-M6\_5\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-1-2 User defined suffixes of the user defined literal operators shall start with underscore followed by one or more letters" name="AUTOSAR-A13\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-2-1 An object shall not be assigned to an overlapping object" name="AUTOSAR-M0\_2\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M4-10-1 NULL shall not be used as an integer value" name="AUTOSAR-M4\_10\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-1-3 User defined literals operators shall only perform conversion of passed parameters" name="AUTOSAR-A13\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-3-3 Pointers to incomplete class types shall not be deleted" name="AUTOSAR-A5\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-3-2 Null pointers shall not be dereferenced" name="AUTOSAR-A5\_3\_2" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A5-3-1 Evaluation of the operand to the typeid operator shall not contain side effects" name="AUTOSAR-A5\_3\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M18-7-1 The signal handling facilities of &lt;csignal> shall not be used" name="AUTOSAR-M18\_7\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A26-5-1 Pseudorandom numbers shall not be generated using std::rand()" name="AUTOSAR-A26\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M4-10-2 Literal zero (0) shall not be used as the null-pointer-constant" name="AUTOSAR-M4\_10\_2" total="1">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="AUTOSAR A2-10-6 A class or enumeration name shall not be hidden by a variable, function or enumerator declaration in the same scope" name="AUTOSAR-A2\_10\_6" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A26-5-2 Random number engines shall not be default-initialized" name="AUTOSAR-A26\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-10-4 The identifier name of a non-member object with static storage duration or static function shall not be reused within a namespace" name="AUTOSAR-A2\_10\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-10-5 An identifier name of a function with static storage duration or a non-member object with external or internal linkage should not be reused" name="AUTOSAR-A2\_10\_5" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A4-5-1 Expressions with type enum or enum class shall not be used as operands to built-in and overloaded operators other than the subscript operator [ ], the assignment operator =, the equality operators == and !=, the unary &amp; operator, and the relational operators &lt;, &lt;=, >, >=" name="AUTOSAR-A4\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-5-3 The std::terminate() function shall not be called implicitly" name="AUTOSAR-A15\_5\_3" total="11">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A15-5-2 Program shall not be abruptly terminated. In particular, an implicit or explicit invocation of std::abort(), std::quick\_exit(), std::\_Exit(), std::terminate() shall not be done" name="AUTOSAR-A15\_5\_2" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-5-1 All user-provided class destructors, deallocation functions, move constructors, move assignment operators and swap functions shall not exit with an exception. A noexcept exception specification shall be added to these functions as appropriate" name="AUTOSAR-A15\_5\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M9-6-4 Named bit-fields with signed integer type shall have a length of more than one bit" name="AUTOSAR-M9\_6\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-2-10 The increment (++) and decrement (--) operators shall not be mixed with other operators in an expression" name="AUTOSAR-M5\_2\_10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-2-12 An identifier with array type passed as a function argument shall not decay to a pointer" name="AUTOSAR-M5\_2\_12" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-2-11 The comma operator, &amp;&amp; operator and the || operator shall not be overloaded" name="AUTOSAR-M5\_2\_11" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A14-7-2 Template specialization shall be declared in the same file (1) as the primary template (2) as a user-defined type, for which the specialization is declared" name="AUTOSAR-A14\_7\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-9-1 Fixed width integer types from &lt;cstdint>, indicating the size and signedness, shall be used in place of the basic numerical types" name="AUTOSAR-A3\_9\_1" total="1">

<Stats authTot="20;" authUrg="0;" total="20"/>

</Category>

<Category desc="AUTOSAR M16-2-3 Include guards shall be provided" name="AUTOSAR-M16\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-1-6 Trivial accessor and mutator functions should be inlined" name="AUTOSAR-A3\_1\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-1-5 A function definition shall only be placed in a class definition if (1) the function is intended to be inlined (2) it is a member function template (3) it is a member function of a class template" name="AUTOSAR-A3\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-1-4 When an array with external linkage is declared, its size shall be stated explicitly" name="AUTOSAR-A3\_1\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-1-3 Implementation files, that are defined locally in the project, should have a file name extension of &quot;.cpp&quot;" name="AUTOSAR-A3\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-1-2 Header files, that are defined locally in the project, shall have a file name extension of one of: &quot;.h&quot;, &quot;.hpp&quot; or &quot;.hxx&quot;" name="AUTOSAR-A3\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-1-1 It shall be possible to include any header file in multiple translation units without violating the One Definition Rule" name="AUTOSAR-A3\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-3-1 Only those characters specified in the C++ Language Standard basic source character set shall be used in the source code" name="AUTOSAR-A2\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M10-1-2 A base class shall only be declared virtual if it is used in a diamond hierarchy" name="AUTOSAR-M10\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M10-1-1 Classes should not be derived from virtual bases" name="AUTOSAR-M10\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M8-0-1 An init-declarator-list or a member-declarator-list shall consist of a single init-declarator or member-declarator respectively" name="AUTOSAR-M8\_0\_1" total="1">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Category>

<Category desc="AUTOSAR M6-4-2 All if ... else if constructs shall be terminated with an else clause" name="AUTOSAR-M6\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-4-1 An if ( condition ) construct shall be followed by a compound statement. The else keyword shall be followed by either a compound statement, or another if statement" name="AUTOSAR-M6\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A17-1-1 Use of the C Standard Library shall be encapsulated and isolated" name="AUTOSAR-A17\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A21-8-1 Arguments to character-handling functions shall be representable as an unsigned char" name="AUTOSAR-A21\_8\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M10-1-3 An accessible base class shall not be both virtual and non-virtual in the same hierarchy" name="AUTOSAR-M10\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-4-6 The final clause of a switch statement shall be the default-clause" name="AUTOSAR-M6\_4\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A25-1-1 Non-static data members or captured values of predicate function objects that are state related to this object's identity shall not be copied" name="AUTOSAR-A25\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-4-5 An unconditional throw or break statement shall terminate every non-empty switch-clause" name="AUTOSAR-M6\_4\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-4-4 A switch-label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" name="AUTOSAR-M6\_4\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-4-3 A switch statement shall be a well-formed switch statement" name="AUTOSAR-M6\_4\_3" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M2-7-1 The character sequence /\* shall not be used within a C-style comment" name="AUTOSAR-M2\_7\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-4-7 The condition of a switch statement shall not have bool type" name="AUTOSAR-M6\_4\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-2-1 dynamic\_cast should not be used" name="AUTOSAR-A5\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-3-2 If a function generates error information, then that error information shall be tested" name="AUTOSAR-M0\_3\_2" total="1">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Category>

<Category desc="AUTOSAR A5-2-5 An array or container shall not be accessed beyond its range" name="AUTOSAR-A5\_2\_5" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-3-1 Minimization of run-time failures shall be ensured by the use of at least one of: (a) static analysis tools/techniques; (b) dynamic analysis tools/techniques; (c) explicit coding of checks to handle run-time faults" name="AUTOSAR-M0\_3\_1" total="10">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A5-2-4 reinterpret\_cast shall not be used" name="AUTOSAR-A5\_2\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-2-3 A cast shall not remove any const or volatile qualification from the type of a pointer or reference" name="AUTOSAR-A5\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-2-2 Traditional C-style casts shall not be used" name="AUTOSAR-A5\_2\_2" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A15-4-5 Checked exceptions that could be thrown from a function shall be specified together with the function declaration and they shall be identical in all function declarations and for all its overriders" name="AUTOSAR-A15\_4\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M17-0-2 The names of standard library macros and objects shall not be reused" name="AUTOSAR-M17\_0\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-4-3 The noexcept specification of a function shall either be identical across all translation units, or identical or more restrictive between a virtual member function and an overrider" name="AUTOSAR-A15\_4\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-4-4 A declaration of non-throwing function shall contain noexcept specification" name="AUTOSAR-A15\_4\_4" total="1">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Category>

<Category desc="AUTOSAR M17-0-3 The names of standard library functions shall not be overridden" name="AUTOSAR-M17\_0\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-4-1 Dynamic exception-specification shall not be used" name="AUTOSAR-A15\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-4-2 If a function is declared to be noexcept, noexcept(true) or noexcept(&lt;true condition>), then it shall not exit with an exception" name="AUTOSAR-A15\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A10-1-1 Class shall not be derived from more than one base class which is not an interface class" name="AUTOSAR-A10\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M17-0-5 The setjmp macro and the longjmp function shall not be used" name="AUTOSAR-M17\_0\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-1-2 All #else, #elif and #endif pre-processor directives shall reside in the same file as the #if or #ifdef directive to which they are related" name="AUTOSAR-M16\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A11-3-1 Friend declarations shall not be used" name="AUTOSAR-A11\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-1-1 The defined pre-processor operator shall only be used in one of the two standard forms" name="AUTOSAR-M16\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-5-2 The address of an object with automatic storage shall not be assigned to another object that may persist after the first object has ceased to exist" name="AUTOSAR-M7\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-5-1 A function shall not return a reference or a pointer to an automatic variable (including parameters), defined within the function" name="AUTOSAR-M7\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M3-2-4 An identifier with external linkage shall have exactly one definition" name="AUTOSAR-M3\_2\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A16-0-1 The pre-processor shall only be used for unconditional and conditional file inclusion and include guards, and using the following directives: (1) #ifndef, (2) #ifdef, (3) #if, (4) #if defined, (5) #elif, (6) #else, (7) #define, (8) #endif, (9) #include" name="AUTOSAR-A16\_0\_1" total="5">

<Stats authTot="18;" authUrg="0;" total="18"/>

</Category>

<Category desc="AUTOSAR M10-2-1 All accessible entity names within a multiple inheritance hierarchy should be unique" name="AUTOSAR-M10\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M8-3-1 Parameters in an overriding virtual function shall either use the same default arguments as the function they override, or else shall not specify any default arguments" name="AUTOSAR-M8\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-1-1 Constructors shall explicitly initialize all virtual base classes, all direct non-virtual base classes and all non-static data members" name="AUTOSAR-A12\_1\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-1-2 Both NSDMI and a non-static member initializer in a constructor shall not be used in the same type" name="AUTOSAR-A12\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-3-4 Each exception explicitly thrown in the code shall have a handler of a compatible type in all call paths that could lead to that point" name="AUTOSAR-M15\_3\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-1-3 If all user-defined constructors of a class initialize data members with constant values that are the same across all constructors, then data members shall be initialized using NSDMI instead" name="AUTOSAR-A12\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-3-7 Where multiple handlers are provided in a single try-catch statement or function-try-block, any ellipsis (catch-all) handler shall occur last" name="AUTOSAR-M15\_3\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-1-4 All constructors that are callable with a single argument of fundamental type shall be declared explicit" name="AUTOSAR-A12\_1\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-3-6 Where multiple handlers are provided in a single try-catch statement or function-try-block for a derived class and some or all of its bases, the handlers shall be ordered most-derived to base class" name="AUTOSAR-M15\_3\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M3-2-1 All declarations of an object or function shall have compatible types" name="AUTOSAR-M3\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A17-6-1 Non-standard entities shall not be added to standard namespaces" name="AUTOSAR-A17\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M3-2-3 A type, object or function that is used in multiple translation units shall be declared in one and only one file" name="AUTOSAR-M3\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-10-1 A pointer to member virtual function shall only be tested for equality with null-pointer-constant" name="AUTOSAR-A5\_10\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M3-2-2 The One Definition Rule shall not be violated" name="AUTOSAR-M3\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-2-6 The operands of a logical &amp;&amp; or || shall be parenthesized if the operands contain binary operators" name="AUTOSAR-A5\_2\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-1-5 Common class initialization for non-constant members shall be done by a delegating constructor" name="AUTOSAR-A12\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-3-1 Exceptions shall be raised only after start-up and before termination" name="AUTOSAR-M15\_3\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A12-1-6 Derived classes that do not need further explicit initialization and require all the constructors from the base class shall use inheriting constructors" name="AUTOSAR-A12\_1\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M15-3-3 Handlers of a function-try-block implementation of a class constructor or destructor shall not reference non-static members from this class or its bases" name="AUTOSAR-M15\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M11-0-1 Member data in non-POD class types shall be private" name="AUTOSAR-M11\_0\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-1-2 Variables shall not be implicitly captured in a lambda expression" name="AUTOSAR-A5\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-1-1 Literal values shall not be used apart from type initialization, otherwise symbolic names shall be used instead" name="AUTOSAR-A5\_1\_1" total="1">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="AUTOSAR A4-10-1 Only nullptr literal shall be used as the null-pointer-constant" name="AUTOSAR-A4\_10\_1" total="2">

<Stats authTot="21;" authUrg="0;" total="21"/>

</Category>

<Category desc="AUTOSAR A5-1-6 Return type of a non-void return type lambda expression should be explicitly specified" name="AUTOSAR-A5\_1\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-1-4 A lambda expression object shall not outlive any of its reference-captured objects" name="AUTOSAR-A5\_1\_4" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-1-3 Parameter list (possibly empty) shall be included in every lambda expression" name="AUTOSAR-A5\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-3-4 Catch-all (ellipsis and std::exception) handlers shall be used only in (a) main, (b) task main functions, (c) in functions that are supposed to isolate independent components and (d) when calling third-party code that uses exceptions not according to AUTOSAR C++14 guidelines" name="AUTOSAR-A15\_3\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-3-5 A class type exception shall be caught by reference or const reference" name="AUTOSAR-A15\_3\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-3-2 If a function throws an exception, it shall be handled when meaningful actions can be taken, otherwise it shall be propagated" name="AUTOSAR-A15\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M2-13-2 Octal constants (other than zero) and octal escape sequences (other than &quot;\0&quot; ) shall not be used" name="AUTOSAR-M2\_13\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-3-3 Main function and a task main function shall catch at least: base class exceptions from all third-party libraries used, std::exception and all otherwise unhandled exceptions" name="AUTOSAR-A15\_3\_3" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A7-6-1 Functions declared with the [[noreturn]] attribute shall not return" name="AUTOSAR-A7\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M2-13-4 Literal suffixes shall be upper case" name="AUTOSAR-M2\_13\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M2-13-3 A &quot;U&quot; suffix shall be applied to all octal or hexadecimal integer literals of unsigned type" name="AUTOSAR-M2\_13\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-19-1 Evaluation of constant unsigned integer expressions shall not lead to wrap-around" name="AUTOSAR-M5\_19\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A14-5-1 A template constructor shall not participate in overload resolution for a single argument of the enclosing class type" name="AUTOSAR-A14\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A10-2-1 Non-virtual member functions shall not be redefined in derived classes" name="AUTOSAR-A10\_2\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-2-1 When declaring function templates, the trailing return type syntax shall be used if the return type depends on the type of parameters" name="AUTOSAR-A8\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A14-5-3 A non-member generic operator shall only be declared in a namespace that does not contain class (struct) type, enum type or union type declarations." name="AUTOSAR-A14\_5\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-6 &quot;forward&quot; parameters declared as T &amp;&amp; shall always be forwarded" name="AUTOSAR-A8\_4\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-0-5 Arguments to a function-like macro shall not contain tokens that look like pre-processing directives" name="AUTOSAR-M16\_0\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-5 &quot;consume&quot; parameters declared as X &amp;&amp; shall always be moved from" name="AUTOSAR-A8\_4\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-8 Output parameters shall not be used" name="AUTOSAR-A8\_4\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-4-1 All usage of assembler shall be documented" name="AUTOSAR-M7\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-7 &quot;in&quot; parameters for &quot;cheap to copy&quot; types shall be passed by value" name="AUTOSAR-A8\_4\_7" total="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR M16-0-2 Macros shall only be #define'd or #undef'd in the global namespace" name="AUTOSAR-M16\_0\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-8-6 Copy and move constructors and copy assignment and move assignment operators shall be declared protected or defined &quot;=delete&quot; in base class" name="AUTOSAR-A12\_8\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-3-2 Static and thread-local objects shall be constant-initialized" name="AUTOSAR-A3\_3\_2" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A8-4-2 All exit paths from a function with non-void return type shall have an explicit return statement with an expression" name="AUTOSAR-A8\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-0-1 #include directives in a file shall only be preceded by other pre-processor directives or comments" name="AUTOSAR-M16\_0\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-4-3 Assembly language shall be encapsulated and isolated" name="AUTOSAR-M7\_4\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-8-7 Assignment operators should be declared with the ref-qualifier &amp;" name="AUTOSAR-A12\_8\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A3-3-1 Objects or functions with external linkage (including members of named namespaces) shall be declared in a header file" name="AUTOSAR-A3\_3\_1" total="2">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Category>

<Category desc="AUTOSAR A8-4-1 Functions shall not be defined using the ellipsis notation" name="AUTOSAR-A8\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M7-4-2 Assembler instructions shall only be introduced using the asm declaration" name="AUTOSAR-M7\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-4 Multiple output values from a function should be returned as a struct or tuple" name="AUTOSAR-A8\_4\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-3 Common ways of passing parameters should be used" name="AUTOSAR-A8\_4\_3" total="2">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Category>

<Category desc="AUTOSAR A7-4-1 The asm declaration shall not be used" name="AUTOSAR-A7\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-5-1 Trigraphs shall not be used" name="AUTOSAR-A2\_5\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A2-5-2 Digraphs shall not be used" name="AUTOSAR-A2\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A8-4-9 &quot;in-out&quot; parameters declared as T &amp; shall be modified" name="AUTOSAR-A8\_4\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-0-8 If the # token appears as the first token on a line, then it shall be immediately followed by a pre-processing token" name="AUTOSAR-M16\_0\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-0-7 Undefined macro identifiers shall not be used in #if or #elif pre-processor directives, except as operands to the defined operator" name="AUTOSAR-M16\_0\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M16-0-6 In the definition of a function-like macro, each instance of a parameter shall be enclosed in parentheses, unless it is used as the operand of # or ##" name="AUTOSAR-M16\_0\_6" total="1">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Category>

<Category desc="AUTOSAR A12-0-2 Bitwise operations and operations that assume data representation in memory shall not be performed on objects" name="AUTOSAR-A12\_0\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M10-3-3 A virtual function shall only be overridden by a pure virtual function if it is itself declared as pure virtual" name="AUTOSAR-M10\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-6-3 The continue statement shall only be used within a well-formed for loop" name="AUTOSAR-M6\_6\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M3-3-2 If a function has internal linkage then all re-declarations shall include the static storage class specifier" name="AUTOSAR-M3\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-6-2 The goto statement shall jump to a label declared later in the same function body" name="AUTOSAR-M6\_6\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-0-1 If a class declares a copy or move operation, or a destructor, either via &quot;=default&quot;, &quot;=delete&quot;, or via a user-provided declaration, then all others of these five special member functions shall be declared as well" name="AUTOSAR-A12\_0\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M6-6-1 Any label referenced by a goto statement shall be declared in the same block, or in a block enclosing the goto statement" name="AUTOSAR-M6\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-8-2 User-defined copy and move assignment operators should use user-defined no-throw swap function" name="AUTOSAR-A12\_8\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-8-3 Moved-from object shall not be read-accessed" name="AUTOSAR-A12\_8\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-8-4 Move constructor shall not initialize its class members and base classes using copy semantics" name="AUTOSAR-A12\_8\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-1-8 Lambda expressions should not be defined inside another lambda expression" name="AUTOSAR-A5\_1\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-8-5 A copy assignment and a move assignment operators shall handle self-assignment" name="AUTOSAR-A12\_8\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-1-7 A lambda shall not be an operand to decltype or typeid" name="AUTOSAR-A5\_1\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M19-3-1 The error indicator errno shall not be used" name="AUTOSAR-M19\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A12-8-1 Move and copy constructors shall move and respectively copy base classes and data members of a class, without any side effects" name="AUTOSAR-A12\_8\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-0-3 The declaration of objects shall contain no more than two levels of pointer indirection" name="AUTOSAR-A5\_0\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-9 An explicit integral conversion shall not change the signedness of the underlying type of a cvalue expression" name="AUTOSAR-M5\_0\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-8-1 The right hand operand of a shift operator shall lie between zero and one less than the width in bits of the underlying type of the left hand operand" name="AUTOSAR-M5\_8\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A0-1-1 A project shall not contain instances of non-volatile variables being given values that are not subsequently used" name="AUTOSAR-A0\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-0-2 The condition of an if-statement and the condition of an iteration statement shall have type bool" name="AUTOSAR-A5\_0\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-8 An explicit integral or floating-point conversion shall not increase the size of the underlying type of a cvalue expression" name="AUTOSAR-M5\_0\_8" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A0-1-2 The value returned by a function having a non-void return type that is not an overloaded operator shall be used" name="AUTOSAR-A0\_1\_2" total="1">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Category>

<Category desc="AUTOSAR A5-0-1 The value of an expression shall be the same under any order of evaluation that the standard permits" name="AUTOSAR-A5\_0\_1" total="7">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-1-4 A project shall not contain non-volatile POD variables having only one use" name="AUTOSAR-M0\_1\_4" total="1">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Category>

<Category desc="AUTOSAR A0-1-3 Every function defined in an anonymous namespace, or static function with internal linkage, or private member function shall be used" name="AUTOSAR-A0\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-1-2 A project shall not contain infeasible paths" name="AUTOSAR-M0\_1\_2" total="29">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-5 There shall be no implicit floating-integral conversions" name="AUTOSAR-M5\_0\_5" total="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="AUTOSAR A27-0-4 C-style strings shall not be used" name="AUTOSAR-A27\_0\_4" total="4">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Category>

<Category desc="AUTOSAR M0-1-3 A project shall not contain unused variables" name="AUTOSAR-M0\_1\_3" total="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR M5-0-4 An implicit integral conversion shall not change the signedness of the underlying type" name="AUTOSAR-M5\_0\_4" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A18-5-2 Non-placement new or delete expressions shall not be used" name="AUTOSAR-A18\_5\_2" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A27-0-3 Alternate input and output operations on a file stream shall not be used without an intervening flush or positioning call" name="AUTOSAR-A27\_0\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-7 There shall be no explicit floating-integral conversions of a cvalue expression" name="AUTOSAR-M5\_0\_7" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-5-1 Functions malloc, calloc, realloc and free shall not be used" name="AUTOSAR-A18\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A27-0-2 A C-style string shall guarantee sufficient space for data and the null terminator" name="AUTOSAR-A27\_0\_2" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A5-0-4 Pointer arithmetic shall not be used with pointers to non-final classes" name="AUTOSAR-A5\_0\_4" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-1-1 A project shall not contain unreachable code" name="AUTOSAR-M0\_1\_1" total="7">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-6 An implicit integral or floating-point conversion shall not reduce the size of the underlying type" name="AUTOSAR-M5\_0\_6" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A13-6-1 Digit sequences separators ' shall only be used as follows: (1) for decimal, every 3 digits, (2) for hexadecimal, every 2 digits, (3) for binary, every 4 digits" name="AUTOSAR-A13\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A27-0-1 Inputs from independent components shall be validated" name="AUTOSAR-A27\_0\_1" total="8">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A0-1-4 There shall be no unused named parameters in non-virtual functions" name="AUTOSAR-A0\_1\_4" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A0-1-5 There shall be no unused named parameters in the set of parameters for a virtual function and all the functions that override it" name="AUTOSAR-A0\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A0-1-6 There should be no unused type declarations" name="AUTOSAR-A0\_1\_6" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A10-3-5 A user-defined assignment operator shall not be virtual" name="AUTOSAR-A10\_3\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A15-2-1 Constructors that are not noexcept shall not be invoked before program startup" name="AUTOSAR-A15\_2\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-5-2 Functions shall not call themselves, either directly or indirectly" name="AUTOSAR-A7\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-18-1 The comma operator shall not be used" name="AUTOSAR-M5\_18\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A7-5-1 A function shall not return a reference or a pointer to a parameter that is passed by reference to const" name="AUTOSAR-A7\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-5-4 If a project has sized or unsized version of operator &quot;delete&quot; globally defined, then both sized and unsized versions shall be defined" name="AUTOSAR-A18\_5\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-5-3 The form of the delete expression shall match the form of the new expression used to allocate the memory" name="AUTOSAR-A18\_5\_3" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-3 A cvalue expression shall not be implicitly converted to a different underlying type" name="AUTOSAR-M5\_0\_3" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A18-5-5 Memory management functions shall ensure the following: (a) deterministic behavior resulting with the existence of worst-case execution time, (b) avoiding memory fragmentation, (c) avoid running out of memory, (d) avoiding mismatched allocations or deallocations, (e) no dependence on non-deterministic calls to kernel" name="AUTOSAR-A18\_5\_5" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M5-0-2 Limited dependence should be placed on C++ operator precedence rules in expressions" name="AUTOSAR-M5\_0\_2" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A10-3-1 Virtual function declaration shall contain exactly one of the three specifiers:" name="AUTOSAR-A10\_3\_1" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A18-5-8 Objects that do not outlive a function shall have automatic storage duration" name="AUTOSAR-A18\_5\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR A10-3-2 Each overriding virtual function shall be declared with the override or final specifier" name="AUTOSAR-A10\_3\_2" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="AUTOSAR A10-3-3 Virtual functions shall not be introduced in a final class" name="AUTOSAR-A10\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-1-8 All functions with void return type shall have external side effect(s)" name="AUTOSAR-M0\_1\_8" total="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="AUTOSAR A18-5-9 Custom implementations of dynamic memory allocation and deallocation functions shall meet the semantic requirements specified in the corresponding &quot;Required behaviour&quot; clause from the C++ Standard" name="AUTOSAR-A18\_5\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="AUTOSAR M0-1-9 There shall be no dead code" name="AUTOSAR-M0\_1\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Flow Analysis" name="BD" total="122">

<Category desc="Incorrect API usage" name="BD-API" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Miscellaneous" name="BD-MISC" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Possible Bugs" name="BD-PB" total="67">

<Stats authTot="4;" authUrg="2;" total="4"/>

</Category>

<Category desc="Resources" name="BD-RES" total="8">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Category>

<Category desc="Security" name="BD-SECURITY" total="19">

<Stats authTot="2;" authUrg="1;" total="2"/>

</Category>

<Category desc="Containers" name="BD-CO" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Threads &amp; Synchronization" name="BD-TRS" total="15">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Code Duplication Detection" name="CDD" total="4">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Category>

<Category desc="SEI CERT C" name="CERT\_C" total="363">

<Category desc="CERT-MSC12\_C (REC) Detect and remove code that has no effect or is never executed" name="CERT\_C-MSC12" total="8">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC11\_C (REC) Incorporate diagnostic tests using assertions" name="CERT\_C-MSC11" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC14\_C (REC) Do not introduce unnecessary platform dependencies" name="CERT\_C-MSC14" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC13\_C (REC) Detect and remove unused values" name="CERT\_C-MSC13" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC15\_C (REC) Do not depend on undefined behavior" name="CERT\_C-MSC15" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-WIN00\_C (REC) Be specific when dynamically loading libraries" name="CERT\_C-WIN00" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FLP03\_C (REC) Detect and handle floating-point errors" name="CERT\_C-FLP03" total="4">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-MSC17\_C (REC) Finish every set of statements associated with a case label with a break statement" name="CERT\_C-MSC17" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FLP02\_C (REC) Avoid using floating-point numbers when precise computation is needed" name="CERT\_C-FLP02" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC19\_C (REC) For functions that return an array, prefer returning an empty array over a null value" name="CERT\_C-MSC19" total="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-FLP06\_C (REC) Convert integers to floating point for floating-point operations" name="CERT\_C-FLP06" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT02\_C (REC) Understand integer conversion rules" name="CERT\_C-INT02" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FLP00\_C (REC) Understand the limitations of floating-point numbers" name="CERT\_C-FLP00" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-API01\_C (REC) Avoid laying out strings in memory directly before sensitive data" name="CERT\_C-API01" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-API02\_C (REC) Functions that read or write to or from an array should take an argument to specify the source or target size" name="CERT\_C-API02" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT08\_C (REC) Verify that all integer values are in range" name="CERT\_C-INT08" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT09\_C (REC) Ensure enumeration constants map to unique values" name="CERT\_C-INT09" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT07\_C (REC) Use only explicitly signed or unsigned char type for numeric values" name="CERT\_C-INT07" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT04\_C (REC) Enforce limits on integer values originating from tainted sources" name="CERT\_C-INT04" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT05\_C (REC) Do not use input functions to convert character data if they cannot handle all possible inputs" name="CERT\_C-INT05" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="CERT-ENV30\_C (RULE) Do not modify the object referenced by the return value of certain functions" name="CERT\_C-ENV30" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC22\_C (REC) Use the setjmp(), longjmp() facility securely" name="CERT\_C-MSC22" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC24\_C (REC) Do not use deprecated or obsolescent functions" name="CERT\_C-MSC24" total="4">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="CERT-INT13\_C (REC) Use bitwise operators only on unsigned operands" name="CERT\_C-INT13" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT12\_C (REC) Do not make assumptions about the type of a plain int bit-field when used in an expression" name="CERT\_C-INT12" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT10\_C (REC) Do not assume a positive remainder when using the % operator" name="CERT\_C-INT10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ENV33\_C (RULE) Do not call system()" name="CERT\_C-ENV33" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ENV34\_C (RULE) Do not store pointers returned by certain functions" name="CERT\_C-ENV34" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT18\_C (REC) Evaluate integer expressions in a larger size before comparing or assigning to that size" name="CERT\_C-INT18" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ENV31\_C (RULE) Do not rely on an environment pointer following an operation that may invalidate it" name="CERT\_C-ENV31" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT15\_C (REC) Use intmax\_t or uintmax\_t for formatted IO on programmer-defined integer types" name="CERT\_C-INT15" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC30\_C (RULE) Do not use the rand() function for generating pseudorandom numbers" name="CERT\_C-MSC30" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ENV32\_C (RULE) All exit handlers must return normally" name="CERT\_C-ENV32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT16\_C (REC) Do not make assumptions about representation of signed integers" name="CERT\_C-INT16" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM00\_C (REC) Allocate and free memory in the same module, at the same level of abstraction" name="CERT\_C-MEM00" total="5">

<Stats authTot="2;" authUrg="2;" total="2"/>

</Category>

<Category desc="CERT-MEM01\_C (REC) Store a new value in pointers immediately after free()" name="CERT\_C-MEM01" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-SIG30\_C (RULE) Call only asynchronous-safe functions within signal handlers" name="CERT\_C-SIG30" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM04\_C (REC) Beware of zero-length allocations" name="CERT\_C-MEM04" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM05\_C (REC) Avoid large stack allocations" name="CERT\_C-MEM05" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-SIG31\_C (RULE) Do not access shared objects in signal handlers" name="CERT\_C-SIG31" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM02\_C (REC) Immediately cast the result of a memory allocation function call into a pointer to the allocated type" name="CERT\_C-MEM02" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-SIG34\_C (RULE) Do not call signal() from within interruptible signal handlers" name="CERT\_C-SIG34" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM03\_C (REC) Clear sensitive information stored in reusable resources" name="CERT\_C-MEM03" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-SIG35\_C (RULE) Do not return from a computational exception signal handler" name="CERT\_C-SIG35" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-PRE31\_C (RULE) Avoid side effects in arguments to unsafe macros" name="CERT\_C-PRE31" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-PRE32\_C (RULE) Do not use preprocessor directives in invocations of function-like macros" name="CERT\_C-PRE32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-PRE30\_C (RULE) Do not create a universal character name through concatenation" name="CERT\_C-PRE30" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ENV02\_C (REC) Beware of multiple environment variables with the same effective name" name="CERT\_C-ENV02" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ENV01\_C (REC) Do not make assumptions about the size of an environment variable" name="CERT\_C-ENV01" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC01\_C (REC) Strive for logical completeness" name="CERT\_C-MSC01" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC04\_C (REC) Use comments consistently and in a readable fashion" name="CERT\_C-MSC04" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC07\_C (REC) Detect and remove dead code" name="CERT\_C-MSC07" total="8">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC09\_C (REC) Character encoding: Use subset of ASCII for safety" name="CERT\_C-MSC09" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT35\_C (RULE) Use correct integer precisions" name="CERT\_C-INT35" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT36\_C (RULE) Converting a pointer to integer or integer to pointer" name="CERT\_C-INT36" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT33\_C (RULE) Ensure that division and remainder operations do not result in divide-by-zero errors" name="CERT\_C-INT33" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-INT34\_C (RULE) Do not shift an expression by a negative number of bits or by greater than or equal to the number of bits that exist in the operand" name="CERT\_C-INT34" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT31\_C (RULE) Ensure that integer conversions do not result in lost or misinterpreted data" name="CERT\_C-INT31" total="15">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT32\_C (RULE) Ensure that operations on signed integers do not result in overflow" name="CERT\_C-INT32" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT30\_C (RULE) Ensure that unsigned integer operations do not wrap" name="CERT\_C-INT30" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO34\_C (RULE) Distinguish between characters read from a file and EOF or WEOF" name="CERT\_C-FIO34" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO32\_C (RULE) Do not perform operations on devices that are only appropriate for files" name="CERT\_C-FIO32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO38\_C (RULE) Do not copy a FILE object" name="CERT\_C-FIO38" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO39\_C (RULE) Do not alternately input and output from a stream without an intervening flush or positioning call" name="CERT\_C-FIO39" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO37\_C (RULE) Do not assume that fgets() or fgetws() returns a nonempty string when successful" name="CERT\_C-FIO37" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP44\_C (RULE) Do not rely on side effects in operands to sizeof, \_Alignof, or \_Generic" name="CERT\_C-EXP44" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP45\_C (RULE) Do not perform assignments in selection statements" name="CERT\_C-EXP45" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP42\_C (RULE) Do not compare padding data" name="CERT\_C-EXP42" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP43\_C (RULE) Avoid undefined behavior when using restrict-qualified pointers" name="CERT\_C-EXP43" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO30\_C (RULE) Exclude user input from format strings" name="CERT\_C-FIO30" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP46\_C (RULE) Do not use a bitwise operator with a Boolean-like operand" name="CERT\_C-EXP46" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP47\_C (RULE) Do not call va\_arg with an argument of the incorrect type" name="CERT\_C-EXP47" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM12\_C (REC) Consider using a goto chain when leaving a function on error when using and releasing resources" name="CERT\_C-MEM12" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-STR34\_C (RULE) Cast characters to unsigned char before converting to larger integer sizes" name="CERT\_C-STR34" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR31\_C (RULE) Guarantee that storage for strings has sufficient space for character data and the null terminator" name="CERT\_C-STR31" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR32\_C (RULE) Do not pass a non-null-terminated character sequence to a library function that expects a string" name="CERT\_C-STR32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR30\_C (RULE) Do not attempt to modify string literals" name="CERT\_C-STR30" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO45\_C (RULE) Avoid TOCTOU race conditions while accessing files" name="CERT\_C-FIO45" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO46\_C (RULE) Do not access a closed file" name="CERT\_C-FIO46" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO44\_C (RULE) Only use values for fsetpos() that are returned from fgetpos()" name="CERT\_C-FIO44" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR37\_C (RULE) Arguments to character-handling functions must be representable as an unsigned char" name="CERT\_C-STR37" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR38\_C (RULE) Do not confuse narrow and wide character strings and functions" name="CERT\_C-STR38" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO47\_C (RULE) Use valid format strings" name="CERT\_C-FIO47" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO41\_C (RULE) Do not call getc(), putc(), getwc(), or putwc() with a stream argument that has side effects" name="CERT\_C-FIO41" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO42\_C (RULE) Close files when they are no longer needed" name="CERT\_C-FIO42" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-FIO40\_C (RULE) Reset strings on fgets() or fgetws() failure" name="CERT\_C-FIO40" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM07\_C (REC) Ensure that the arguments to calloc(), when multiplied, do not wrap" name="CERT\_C-MEM07" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC32\_C (RULE) Properly seed pseudorandom number generators" name="CERT\_C-MSC32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC33\_C (RULE) Do not pass invalid data to the asctime() function" name="CERT\_C-MSC33" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC38\_C (RULE) Do not treat a predefined identifier as an object if it might only be implemented as a macro" name="CERT\_C-MSC38" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC37\_C (RULE) Ensure that control never reaches the end of a non-void function" name="CERT\_C-MSC37" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC39\_C (RULE) Do not call va\_arg() on a va\_list that has an indeterminate value" name="CERT\_C-MSC39" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP30\_C (RULE) Do not depend on the order of evaluation for side effects" name="CERT\_C-EXP30" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON40\_C (RULE) Do not refer to an atomic variable twice in an expression" name="CERT\_C-CON40" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP20\_C (REC) Perform explicit tests to determine success, true and false, and equality" name="CERT\_C-EXP20" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON43\_C (RULE) Do not allow data races in multithreaded code" name="CERT\_C-CON43" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON41\_C (RULE) Wrap functions that can fail spuriously in a loop" name="CERT\_C-CON41" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC41\_C (RULE) Never hard code sensitive information" name="CERT\_C-MSC41" total="1">

<Stats authTot="3;" authUrg="2;" total="3"/>

</Category>

<Category desc="CERT-MSC40\_C (RULE) Do not violate constraints" name="CERT\_C-MSC40" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM33\_C (RULE) Allocate and copy structures containing a flexible array member dynamically" name="CERT\_C-MEM33" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM34\_C (RULE) Only free memory allocated dynamically" name="CERT\_C-MEM34" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM31\_C (RULE) Free dynamically allocated memory when no longer needed" name="CERT\_C-MEM31" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-MEM35\_C (RULE) Allocate sufficient memory for an object" name="CERT\_C-MEM35" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-SIG01\_C (REC) Understand implementation-specific details regarding signal handler persistence" name="CERT\_C-SIG01" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM36\_C (RULE) Do not modify the alignment of objects by calling realloc()" name="CERT\_C-MEM36" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-SIG00\_C (REC) Mask signals handled by noninterruptible signal handlers" name="CERT\_C-SIG00" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-API00\_C (REC) Functions should validate their parameters" name="CERT\_C-API00" total="1">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Category>

<Category desc="CERT-FIO24\_C (REC) Do not open a file that is already open" name="CERT\_C-FIO24" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-SIG02\_C (REC) Avoid using signals to implement normal functionality" name="CERT\_C-SIG02" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO21\_C (REC) Do not create temporary files in shared directories" name="CERT\_C-FIO21" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO22\_C (REC) Close files before spawning processes" name="CERT\_C-FIO22" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-EXP40\_C (RULE) Do not modify constant objects" name="CERT\_C-EXP40" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM30\_C (RULE) Do not access freed memory" name="CERT\_C-MEM30" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP33\_C (RULE) Do not read uninitialized memory" name="CERT\_C-EXP33" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP34\_C (RULE) Do not dereference null pointers" name="CERT\_C-EXP34" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-EXP32\_C (RULE) Do not access a volatile object through a nonvolatile reference" name="CERT\_C-EXP32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON32\_C (RULE) Prevent data races when accessing bit-fields from multiple threads" name="CERT\_C-CON32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP37\_C (RULE) Call functions with the correct number and type of arguments" name="CERT\_C-EXP37" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON33\_C (RULE) Avoid race conditions when using library functions" name="CERT\_C-CON33" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON30\_C (RULE) Clean up thread-specific storage" name="CERT\_C-CON30" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-EXP35\_C (RULE) Do not modify objects with temporary lifetime" name="CERT\_C-EXP35" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON31\_C (RULE) Do not destroy a mutex while it is locked" name="CERT\_C-CON31" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP36\_C (RULE) Do not cast pointers into more strictly aligned pointer types" name="CERT\_C-EXP36" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON36\_C (RULE) Wrap functions that can spuriously wake up in a loop" name="CERT\_C-CON36" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON37\_C (RULE) Do not call signal() in a multithreaded program" name="CERT\_C-CON37" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON34\_C (RULE) Declare objects shared between threads with appropriate storage durations" name="CERT\_C-CON34" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP39\_C (RULE) Do not access a variable through a pointer of an incompatible type" name="CERT\_C-EXP39" total="6">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="CERT-CON35\_C (RULE) Avoid deadlock by locking in a predefined order" name="CERT\_C-CON35" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON38\_C (RULE) Preserve thread safety and liveness when using condition variables" name="CERT\_C-CON38" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON39\_C (RULE) Do not join or detach a thread that was previously joined or detached" name="CERT\_C-CON39" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL22\_C (REC) Use volatile for data that cannot be cached" name="CERT\_C-DCL22" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL20\_C (REC) Explicitly specify void when a function accepts no arguments" name="CERT\_C-DCL20" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP00\_C (REC) Use parentheses for precedence of operation" name="CERT\_C-EXP00" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP05\_C (REC) Do not cast away a const qualification" name="CERT\_C-EXP05" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP02\_C (REC) Be aware of the short-circuit behavior of the logical AND and OR operators" name="CERT\_C-EXP02" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP08\_C (REC) Ensure pointer arithmetic is used correctly" name="CERT\_C-EXP08" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL19\_C (REC) Minimize the scope of variables and functions" name="CERT\_C-DCL19" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL18\_C (REC) Do not begin integer constants with 0 when specifying a decimal value" name="CERT\_C-DCL18" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL12\_C (REC) Implement abstract data types using opaque types" name="CERT\_C-DCL12" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO01\_C (REC) Be careful using functions that use file names for identification" name="CERT\_C-FIO01" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL11\_C (REC) Understand the type issues associated with variadic functions" name="CERT\_C-DCL11" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL10\_C (REC) Maintain the contract between the writer and caller of variadic functions" name="CERT\_C-DCL10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL16\_C (REC) Use &quot;L,&quot; not &quot;l,&quot; to indicate a long value" name="CERT\_C-DCL16" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL15\_C (REC) Declare file-scope objects or functions that do not need external linkage as static" name="CERT\_C-DCL15" total="1">

<Stats authTot="16;" authUrg="0;" total="16"/>

</Category>

<Category desc="CERT-DCL13\_C (REC) Declare function parameters that are pointers to values not changed by the function as const" name="CERT\_C-DCL13" total="1">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="CERT-EXP12\_C (REC) Do not ignore values returned by functions" name="CERT\_C-EXP12" total="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Category>

<Category desc="CERT-EXP10\_C (REC) Do not depend on the order of evaluation of subexpressions or the order in which side effects take place" name="CERT\_C-EXP10" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP15\_C (REC) Do not place a semicolon on the same line as an if, for, or while statement" name="CERT\_C-EXP15" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP16\_C (REC) Do not compare function pointers to constant values" name="CERT\_C-EXP16" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP14\_C (REC) Beware of integer promotion when performing bitwise operations on integer types smaller than int" name="CERT\_C-EXP14" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP19\_C (REC) Use braces for the body of an if, for, or while statement" name="CERT\_C-EXP19" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR00\_C (REC) Represent characters using an appropriate type" name="CERT\_C-STR00" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR09\_C (REC) Don't assume numeric values for expressions with type plain character" name="CERT\_C-STR09" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR07\_C (REC) Use the bounds-checking interfaces for string manipulation" name="CERT\_C-STR07" total="1">

<Stats authTot="2;" authUrg="1;" total="2"/>

</Category>

<Category desc="CERT-STR04\_C (REC) Use plain char for characters in the basic character set" name="CERT\_C-STR04" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR05\_C (REC) Use pointers to const when referring to string literals" name="CERT\_C-STR05" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR02\_C (REC) Sanitize data passed to complex subsystems" name="CERT\_C-STR02" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR03\_C (REC) Do not inadvertently truncate a string" name="CERT\_C-STR03" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL41\_C (RULE) Do not declare variables inside a switch statement before the first case label" name="CERT\_C-DCL41" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL40\_C (RULE) Do not create incompatible declarations of the same function or object" name="CERT\_C-DCL40" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON01\_C (REC) Acquire and release synchronization primitives in the same module, at the same level of abstraction" name="CERT\_C-CON01" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-CON02\_C (REC) Do not use volatile as a synchronization primitive" name="CERT\_C-CON02" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON05\_C (REC) Do not perform operations that can block while holding a lock" name="CERT\_C-CON05" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR11\_C (REC) Do not specify the bound of a character array initialized with a string literal" name="CERT\_C-STR11" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL39\_C (RULE) Avoid information leakage when passing a structure across a trust boundary" name="CERT\_C-DCL39" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR10\_C (REC) Do not concatenate different type of string literals" name="CERT\_C-STR10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL31\_C (RULE) Declare identifiers before using them" name="CERT\_C-DCL31" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL38\_C (RULE) Use the correct syntax when declaring a flexible array member" name="CERT\_C-DCL38" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL37\_C (RULE) Do not declare or define a reserved identifier" name="CERT\_C-DCL37" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL36\_C (RULE) Do not declare an identifier with conflicting linkage classifications" name="CERT\_C-DCL36" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL30\_C (RULE) Declare objects with appropriate storage durations" name="CERT\_C-DCL30" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ARR01\_C (REC) Do not apply the sizeof operator to a pointer when taking the size of an array" name="CERT\_C-ARR01" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ARR02\_C (REC) Explicitly specify array bounds, even if implicitly defined by an initializer" name="CERT\_C-ARR02" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-PRE02\_C (REC) Macro replacement lists should be parenthesized" name="CERT\_C-PRE02" total="1">

<Stats authTot="8;" authUrg="6;" total="8"/>

</Category>

<Category desc="CERT-PRE00\_C (REC) Prefer inline or static functions to function-like macros" name="CERT\_C-PRE00" total="1">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Category>

<Category desc="CERT-PRE01\_C (REC) Use parentheses within macros around parameter names" name="CERT\_C-PRE01" total="1">

<Stats authTot="8;" authUrg="2;" total="8"/>

</Category>

<Category desc="CERT-PRE06\_C (REC) Enclose header files in an include guard" name="CERT\_C-PRE06" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-PRE07\_C (REC) Avoid using repeated question marks" name="CERT\_C-PRE07" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS54\_C (RULE) Detect and handle POSIX library errors" name="CERT\_C-POS54" total="3">

<Stats authTot="18;" authUrg="12;" total="18"/>

</Category>

<Category desc="CERT-ERR30\_C (RULE) Set errno to zero before calling a library function known to set errno, and check errno only after the function returns a value indicating failure" name="CERT\_C-ERR30" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR32\_C (RULE) Do not rely on indeterminate values of errno" name="CERT\_C-ERR32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR33\_C (RULE) Detect and handle standard library errors" name="CERT\_C-ERR33" total="4">

<Stats authTot="18;" authUrg="12;" total="18"/>

</Category>

<Category desc="CERT-ERR34\_C (RULE) Detect errors when converting a string to a number" name="CERT\_C-ERR34" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="CERT-POS51\_C (RULE) Avoid deadlock with POSIX threads by locking in predefined order" name="CERT\_C-POS51" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS50\_C (RULE) Declare objects shared between POSIX threads with appropriate storage durations" name="CERT\_C-POS50" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS53\_C (RULE) Do not use more than one mutex for concurrent waiting operations on a condition variable" name="CERT\_C-POS53" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS52\_C (RULE) Do not perform operations that can block while holding a POSIX lock" name="CERT\_C-POS52" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR06\_C (REC) Understand the termination behavior of assert() and abort()" name="CERT\_C-ERR06" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR07\_C (REC) Prefer functions that support error checking over equivalent functions that don't" name="CERT\_C-ERR07" total="2">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Category>

<Category desc="CERT-DCL06\_C (REC) Use meaningful symbolic constants to represent literal values" name="CERT\_C-DCL06" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL01\_C (REC) Do not reuse variable names in subscopes" name="CERT\_C-DCL01" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL00\_C (REC) Const-qualify immutable objects" name="CERT\_C-DCL00" total="1">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Category>

<Category desc="CERT-DCL05\_C (REC) Use typedefs of non-pointer types only" name="CERT\_C-DCL05" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL04\_C (REC) Do not declare more than one variable per declaration" name="CERT\_C-DCL04" total="1">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Category>

<Category desc="CERT-DCL02\_C (REC) Use visually distinct identifiers" name="CERT\_C-DCL02" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS44\_C (RULE) Do not use signals to terminate threads" name="CERT\_C-POS44" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS48\_C (RULE) Do not unlock or destroy another POSIX thread's mutex" name="CERT\_C-POS48" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS47\_C (RULE) Do not use threads that can be canceled asynchronously" name="CERT\_C-POS47" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS49\_C (RULE) When data must be accessed by multiple threads, provide a mutex and guarantee no adjacent data is also accessed" name="CERT\_C-POS49" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-WIN30\_C (RULE) Properly pair allocation and deallocation functions" name="CERT\_C-WIN30" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-ERR01\_C (REC) Use ferror() rather than errno to check for FILE stream errors" name="CERT\_C-ERR01" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR02\_C (REC) Avoid in-band error indicators" name="CERT\_C-ERR02" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="CERT-ERR04\_C (REC) Choose an appropriate termination strategy" name="CERT\_C-ERR04" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR05\_C (REC) Application-independent code should provide error detection without dictating error handling" name="CERT\_C-ERR05" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FLP37\_C (RULE) Do not use object representations to compare floating-point values" name="CERT\_C-FLP37" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FLP36\_C (RULE) Preserve precision when converting integral values to floating-point type" name="CERT\_C-FLP36" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FLP34\_C (RULE) Ensure that floating-point conversions are within range of the new type" name="CERT\_C-FLP34" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS33\_C (RULE) Do not use vfork()" name="CERT\_C-POS33" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ARR32\_C (RULE) Ensure size arguments for variable length arrays are in a valid range" name="CERT\_C-ARR32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS35\_C (RULE) Avoid race conditions while checking for the existence of a symbolic link" name="CERT\_C-POS35" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS34\_C (RULE) Do not call putenv() with a pointer to an automatic variable as the argument" name="CERT\_C-POS34" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ARR30\_C (RULE) Do not form or use out-of-bounds pointers or array subscripts" name="CERT\_C-ARR30" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS37\_C (RULE) Ensure that privilege relinquishment is successful" name="CERT\_C-POS37" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FLP32\_C (RULE) Prevent or detect domain and range errors in math functions" name="CERT\_C-FLP32" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS36\_C (RULE) Observe correct revocation order while relinquishing privileges" name="CERT\_C-POS36" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS39\_C (RULE) Use the correct byte ordering when transferring data between systems" name="CERT\_C-POS39" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FLP30\_C (RULE) Do not use floating-point variables as loop counters" name="CERT\_C-FLP30" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS38\_C (RULE) Beware of race conditions when using fork and file descriptors" name="CERT\_C-POS38" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ARR39\_C (RULE) Do not add or subtract a scaled integer to a pointer" name="CERT\_C-ARR39" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ARR38\_C (RULE) Guarantee that library functions do not form invalid pointers" name="CERT\_C-ARR38" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ARR37\_C (RULE) Do not add or subtract an integer to a pointer to a non-array object" name="CERT\_C-ARR37" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ARR36\_C (RULE) Do not subtract or compare two pointers that do not refer to the same array" name="CERT\_C-ARR36" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-POS30\_C (RULE) Use the readlink() function properly" name="CERT\_C-POS30" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="SEI CERT C++" name="CERT\_CPP" total="143">

<Category desc="CERT-DCL55\_CPP (RULE) Avoid information leakage when passing a class object across a trust boundary" name="CERT\_CPP-DCL55" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR52\_CPP (RULE) Do not use setjmp() or longjmp()" name="CERT\_CPP-ERR52" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL56\_CPP (RULE) Avoid cycles during initialization of static objects" name="CERT\_CPP-DCL56" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR51\_CPP (RULE) Handle all exceptions" name="CERT\_CPP-ERR51" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON56\_CPP (RULE) Do not speculatively lock a non-recursive mutex that is already owned by the calling thread" name="CERT\_CPP-CON56" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL53\_CPP (RULE) Do not write syntactically ambiguous declarations" name="CERT\_CPP-DCL53" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR50\_CPP (RULE) Do not abruptly terminate the program" name="CERT\_CPP-ERR50" total="14">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="CERT-DCL54\_CPP (RULE) Overload allocation and deallocation functions as a pair in the same scope" name="CERT\_CPP-DCL54" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL59\_CPP (RULE) Do not define an unnamed namespace in a header file" name="CERT\_CPP-DCL59" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM51\_CPP (RULE) Properly deallocate dynamically allocated resources" name="CERT\_CPP-MEM51" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM52\_CPP (RULE) Detect and handle memory allocation errors" name="CERT\_CPP-MEM52" total="2">

<Stats authTot="2;" authUrg="1;" total="2"/>

</Category>

<Category desc="CERT-DCL57\_CPP (RULE) Do not let exceptions escape from destructors or deallocation functions" name="CERT\_CPP-DCL57" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL58\_CPP (RULE) Do not modify the standard namespaces" name="CERT\_CPP-DCL58" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM50\_CPP (RULE) Do not access freed memory" name="CERT\_CPP-MEM50" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON50\_CPP (RULE) Do not destroy a mutex while it is locked" name="CERT\_CPP-CON50" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM55\_CPP (RULE) Honor replacement dynamic storage management requirements" name="CERT\_CPP-MEM55" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON51\_CPP (RULE) Ensure actively held locks are released on exceptional conditions" name="CERT\_CPP-CON51" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR59\_CPP (RULE) Do not throw an exception across execution boundaries" name="CERT\_CPP-ERR59" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM56\_CPP (RULE) Do not store an already-owned pointer value in an unrelated smart pointer" name="CERT\_CPP-MEM56" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR58\_CPP (RULE) Handle all exceptions thrown before main() begins executing" name="CERT\_CPP-ERR58" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-MEM53\_CPP (RULE) Explicitly construct and destruct objects when manually managing object lifetime" name="CERT\_CPP-MEM53" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR57\_CPP (RULE) Do not leak resources when handling exceptions" name="CERT\_CPP-ERR57" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-MEM54\_CPP (RULE) Provide placement new with properly aligned pointers to sufficient storage capacity" name="CERT\_CPP-MEM54" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON54\_CPP (RULE) Wrap functions that can spuriously wake up in a loop" name="CERT\_CPP-CON54" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR56\_CPP (RULE) Guarantee exception safety" name="CERT\_CPP-ERR56" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON55\_CPP (RULE) Preserve thread safety and liveness when using condition variables" name="CERT\_CPP-CON55" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR55\_CPP (RULE) Honor exception specifications" name="CERT\_CPP-ERR55" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON52\_CPP (RULE) Prevent data races when accessing bit-fields from multiple threads" name="CERT\_CPP-CON52" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR54\_CPP (RULE) Catch handlers should order their parameter types from most derived to least derived" name="CERT\_CPP-ERR54" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MEM57\_CPP (RULE) Avoid using default operator new for over-aligned types" name="CERT\_CPP-MEM57" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CON53\_CPP (RULE) Avoid deadlock by locking in a predefined order" name="CERT\_CPP-CON53" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR53\_CPP (RULE) Do not reference base classes or class data members in a constructor or destructor function-try-block handler" name="CERT\_CPP-ERR53" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR53\_CPP (RULE) Range check element access" name="CERT\_CPP-STR53" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL51\_CPP (RULE) Do not declare or define a reserved identifier" name="CERT\_CPP-DCL51" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR50\_CPP (RULE) Guarantee that storage for strings has sufficient space for character data and the null terminator" name="CERT\_CPP-STR50" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL52\_CPP (RULE) Never qualify a reference type with const or volatile" name="CERT\_CPP-DCL52" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR52\_CPP (RULE) Use valid references, pointers, and iterators to reference elements of a basic\_string" name="CERT\_CPP-STR52" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL50\_CPP (RULE) Do not define a C-style variadic function" name="CERT\_CPP-DCL50" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-STR51\_CPP (RULE) Do not attempt to create a std::string from a null pointer" name="CERT\_CPP-STR51" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-OOP53\_CPP (RULE) Write constructor member initializers in the canonical order" name="CERT\_CPP-OOP53" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-OOP52\_CPP (RULE) Do not delete a polymorphic object without a virtual destructor" name="CERT\_CPP-OOP52" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-OOP51\_CPP (RULE) Do not slice derived objects" name="CERT\_CPP-OOP51" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-OOP50\_CPP (RULE) Do not invoke virtual functions from constructors or destructors" name="CERT\_CPP-OOP50" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-OOP57\_CPP (RULE) Prefer special member functions and overloaded operators to C Standard Library functions" name="CERT\_CPP-OOP57" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC50\_CPP (RULE) Do not use std::rand() for generating pseudorandom numbers" name="CERT\_CPP-MSC50" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-OOP56\_CPP (RULE) Honor replacement handler requirements" name="CERT\_CPP-OOP56" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC51\_CPP (RULE) Ensure your random number generator is properly seeded" name="CERT\_CPP-MSC51" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-OOP55\_CPP (RULE) Do not use pointer-to-member operators to access nonexistent members" name="CERT\_CPP-OOP55" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC52\_CPP (RULE) Value-returning functions must return a value from all exit paths" name="CERT\_CPP-MSC52" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-OOP54\_CPP (RULE) Gracefully handle self-copy assignment" name="CERT\_CPP-OOP54" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-OOP58\_CPP (RULE) Copy operations must not mutate the source object" name="CERT\_CPP-OOP58" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-FIO51\_CPP (RULE) Close files when they are no longer needed" name="CERT\_CPP-FIO51" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CERT-FIO50\_CPP (RULE) Do not alternately input and output from a file stream without an intervening positioning call" name="CERT\_CPP-FIO50" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-DCL60\_CPP (RULE) Obey the one-definition rule" name="CERT\_CPP-DCL60" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP55\_CPP (RULE) Do not access a cv-qualified object through a cv-unqualified type" name="CERT\_CPP-EXP55" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP56\_CPP (RULE) Do not call a function with a mismatched language linkage" name="CERT\_CPP-EXP56" total="1">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Category>

<Category desc="CERT-EXP53\_CPP (RULE) Do not read uninitialized memory" name="CERT\_CPP-EXP53" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP54\_CPP (RULE) Do not access an object outside of its lifetime" name="CERT\_CPP-EXP54" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP59\_CPP (RULE) Use offsetof() on valid types and members" name="CERT\_CPP-EXP59" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP57\_CPP (RULE) Do not cast or delete pointers to incomplete classes" name="CERT\_CPP-EXP57" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP58\_CPP (RULE) Pass an object of the correct type to va\_start" name="CERT\_CPP-EXP58" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP62\_CPP (RULE) Do not access the bits of an object representation that are not part of the object's value representation" name="CERT\_CPP-EXP62" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP63\_CPP (RULE) Do not rely on the value of a moved-from object" name="CERT\_CPP-EXP63" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP60\_CPP (RULE) Do not pass a nonstandard-layout type object across execution boundaries" name="CERT\_CPP-EXP60" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP61\_CPP (RULE) A lambda object must not outlive any of its reference captured objects" name="CERT\_CPP-EXP61" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC53\_CPP (RULE) Do not return from a function declared [[noreturn]]" name="CERT\_CPP-MSC53" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-MSC54\_CPP (RULE) A signal handler must be a plain old function" name="CERT\_CPP-MSC54" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR62\_CPP (RULE) Detect errors when converting a string to a number" name="CERT\_CPP-ERR62" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="CERT-ERR61\_CPP (RULE) Catch exceptions by lvalue reference" name="CERT\_CPP-ERR61" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-ERR60\_CPP (RULE) Exception objects must be nothrow copy constructible" name="CERT\_CPP-ERR60" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR57\_CPP (RULE) Provide a valid ordering predicate" name="CERT\_CPP-CTR57" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR58\_CPP (RULE) Predicate function objects should not be mutable" name="CERT\_CPP-CTR58" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR53\_CPP (RULE) Use valid iterator ranges" name="CERT\_CPP-CTR53" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR54\_CPP (RULE) Do not subtract iterators that do not refer to the same container" name="CERT\_CPP-CTR54" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR55\_CPP (RULE) Do not use an additive operator on an iterator if the result would overflow" name="CERT\_CPP-CTR55" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR56\_CPP (RULE) Do not use pointer arithmetic on polymorphic objects" name="CERT\_CPP-CTR56" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR50\_CPP (RULE) Guarantee that container indices and iterators are within the valid range" name="CERT\_CPP-CTR50" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR51\_CPP (RULE) Use valid references, pointers, and iterators to reference elements of a container" name="CERT\_CPP-CTR51" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-CTR52\_CPP (RULE) Guarantee that library functions do not overflow" name="CERT\_CPP-CTR52" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-INT50\_CPP (RULE) Do not cast to an out-of-range enumeration value" name="CERT\_CPP-INT50" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP51\_CPP (RULE) Do not delete an array through a pointer of the incorrect type" name="CERT\_CPP-EXP51" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP52\_CPP (RULE) Do not rely on side effects in unevaluated operands" name="CERT\_CPP-EXP52" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CERT-EXP50\_CPP (RULE) Do not depend on the order of evaluation for side effects" name="CERT\_CPP-EXP50" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Coding Conventions" name="CODSTA" total="441">

<Category desc="Coding Conventions for C++" name="CODSTA-CPP" total="112">

<Stats authTot="116;" authUrg="1;" total="116"/>

</Category>

<Category desc="Coding Conventions for Modern C++" name="CODSTA-MCPP" total="68">

<Stats authTot="56;" authUrg="0;" total="56"/>

</Category>

<Stats authTot="158;" authUrg="0;" total="158"/>

</Category>

<Category desc="Comments" name="COMMENT" total="16">

<Stats authTot="126;" authUrg="0;" total="126"/>

</Category>

<Category desc="Common Weakness Enumeration" name="CWE" total="83">

<Category desc="CWE-532 Inclusion of Sensitive Information in Log Files" name="CWE-532" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-772 Missing Release of Resource after Effective Lifetime" name="CWE-772" total="2">

<Stats authTot="2;" authUrg="1;" total="2"/>

</Category>

<Category desc="CWE-476 NULL Pointer Dereference" name="CWE-476" total="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CWE-770 Allocation of Resources Without Limits or Throttling" name="CWE-770" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CWE-190 Integer Overflow or Wraparound" name="CWE-190" total="7">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-78 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')" name="CWE-78" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-617 Reachable Assertion" name="CWE-617" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-835 Loop with Unreachable Exit Condition ('Infinite Loop')" name="CWE-835" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-415 Double Free" name="CWE-415" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer" name="CWE-119" total="11">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-416 Use After Free" name="CWE-416" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-611 Improper Restriction of XML External Entity Reference" name="CWE-611" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-732 Incorrect Permission Assignment for Critical Resource" name="CWE-732" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-798 Use of Hard-coded Credentials" name="CWE-798" total="1">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="CWE-125 Out-of-bounds Read" name="CWE-125" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-400 Uncontrolled Resource Consumption" name="CWE-400" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-269 Improper Privilege Management" name="CWE-269" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-863 Incorrect Authorization" name="CWE-863" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-200 Information Exposure" name="CWE-200" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-287 Improper Authentication" name="CWE-287" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-362 Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')" name="CWE-362" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-20 Improper Input Validation" name="CWE-20" total="10">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')" name="CWE-22" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')" name="CWE-89" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-704 Incorrect Type Conversion or Cast" name="CWE-704" total="12">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-426 Untrusted Search Path" name="CWE-426" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-787 Out-of-bounds Write" name="CWE-787" total="7">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="CWE-326 Inadequate Encryption Strength" name="CWE-326" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Exceptions" name="EXCEPT" total="26">

<Stats authTot="5;" authUrg="0;" total="5"/>

</Category>

<Category desc="Formatting" name="FORMAT" total="51">

<Stats authTot="274;" authUrg="0;" total="274"/>

</Category>

<Category desc="Global Static Analysis" name="GLOBAL" total="20">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Category>

<Category desc="High Integrity C++" name="HICPP" total="234">

<Category desc="HIC++ 7.2.2 Initialize none, the first only or all enumerators in an enumeration" name="HICPP-7\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 2.1.1 Do not use tab characters in source files" name="HICPP-2\_1\_1" total="1">

<Stats authTot="160;" authUrg="0;" total="160"/>

</Category>

<Category desc="HIC++ 2.5.3 Use nullptr for the null pointer constant" name="HICPP-2\_5\_3" total="1">

<Stats authTot="21;" authUrg="0;" total="21"/>

</Category>

<Category desc="HIC++ 15.1.1 Only use instances of std::exception for exceptions" name="HICPP-15\_1\_1" total="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="HIC++ 2.5.2 Do not use octal constants (other than zero)" name="HICPP-2\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 2.5.1 Do not concatenate strings with different encoding prefixes" name="HICPP-2\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.4.4 Write members in an initialization list in the order in which they are declared" name="HICPP-12\_4\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.2.1 Use an explicit enumeration base and ensure that it is large enough to store all enumerators" name="HICPP-7\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.4.5 Use delegating constructors to reduce code duplication" name="HICPP-12\_4\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 3.3.1 Do not use variables with static storage duration" name="HICPP-3\_3\_1" total="1">

<Stats authTot="7;" authUrg="0;" total="7"/>

</Category>

<Category desc="HIC++ 17.5.1 Do not ignore the result of std::remove, std::remove if or std::unique" name="HICPP-17\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 13.2.5 Implement a minimal set of operators and use them to implement all other related operators" name="HICPP-13\_2\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 4.1.1 Ensure that a function argument does not undergo an array-to-pointer conversion" name="HICPP-4\_1\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 8.4.2 Ensure that a braced aggregate initializer matches the layout of the aggregate object" name="HICPP-8\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 8.4.1 Do not access an invalid object or an object with indeterminate value" name="HICPP-8\_4\_1" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 13.1.2 If a member of a set of callable functions includes a universal reference parameter, ensure that one appears in the same position for all other members" name="HICPP-13\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.3.1 Within the scope of a lock, ensure that no static path results in a lock of the same mutex" name="HICPP-18\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.3.2 Ensure that order of nesting of locks in a project forms a DAG" name="HICPP-18\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.3.5 Do not access the members of std::mutex directly" name="HICPP-18\_3\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.3.6 Do not use relaxed atomics" name="HICPP-18\_3\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.3.3 Ensure that the form of delete matches the form of new used to allocate the memory" name="HICPP-5\_3\_3" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.3.3 Do not use std::recursive mutex" name="HICPP-18\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.7.2 Ensure that a pointer to member that is a virtual function is only compared (==) with nullptr" name="HICPP-5\_7\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 13.1.1 Ensure that all overloads of a function are visible from where it is called" name="HICPP-13\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.3.4 Only use std::unique lock when std::lock guard cannot be used" name="HICPP-18\_3\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.7.1 Do not write code that expects floating point calculations to yield exact results" name="HICPP-5\_7\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.3.2 Allocate memory using new and release it using delete" name="HICPP-5\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.3.1 Do not apply unary minus to operands of unsigned type" name="HICPP-5\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 9.2.1 Declare bit-fields with an explicitly unsigned integral or enumeration type" name="HICPP-9\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.3.1 Correctly declare overloads for operator new and delete" name="HICPP-12\_3\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.1.4 Ensure that a switch statement has at least two case labels, distinct from the default label" name="HICPP-6\_1\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 11.1.1 Declare all data members private" name="HICPP-11\_1\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.1.3 Ensure that a non-empty case statement block does not fall through to the next label" name="HICPP-6\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.1.2 Explicitly cover all paths through multi-way selection statements" name="HICPP-6\_1\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.1.1 Enclose the body of a selection or an iteration statement in a compound statement" name="HICPP-6\_1\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.3.1 Do not use using directives" name="HICPP-7\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 15.2.1 Do not throw an exception from a destructor" name="HICPP-15\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 2.4.1 Ensure that each identifier is distinct from any other visible identifier" name="HICPP-2\_4\_1" total="1">

<Stats authTot="13;" authUrg="0;" total="13"/>

</Category>

<Category desc="HIC++ 12.5.7 Declare assignment operators with the ref-qualifier &amp;" name="HICPP-12\_5\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 10.3.1 Ensure that a derived class has at most one base class which is not an interface class" name="HICPP-10\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.5.8 Make the copy assignment operator of an abstract class protected or define it =delete" name="HICPP-12\_5\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.5.6 Use an atomic, non-throwing swap operation to implement the copy and move assignment operators" name="HICPP-12\_5\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.5.3 Ensure that a user defined move/copy constructor only moves/copies base and member objects" name="HICPP-12\_5\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.5.4 Declare noexcept the move constructor and move assignment operator" name="HICPP-12\_5\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 8.1.1 Do not use multiple levels of pointer indirection" name="HICPP-8\_1\_1" total="1">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="HIC++ 3.2.1 Do not declare functions at block scope" name="HICPP-3\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 17.4.1 Use const container calls when result is immediately converted to a const iterator" name="HICPP-17\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 17.4.2 Use API calls that construct objects in place" name="HICPP-17\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 4.4.1 Do not convert floating values to integral types except through use of standard library functions" name="HICPP-4\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 13.2.1 Do not overload operators with special semantics" name="HICPP-13\_2\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.2.2 Synchronize access to data shared between threads using a single lock" name="HICPP-18\_2\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 13.2.2 Ensure that the return type of an overloaded binary operator matches the built-in counterparts" name="HICPP-13\_2\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.2.3 Do not share volatile data between threads" name="HICPP-18\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 13.2.3 Declare binary arithmetic and bitwise operators as non-members" name="HICPP-13\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 13.2.4 When overloading the subscript operator (operator[]) implement both const and non-const versions" name="HICPP-13\_2\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.2.1 Use high integrity::thread in place of std::thread" name="HICPP-18\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.6.1 Do not use bitwise operators with signed operands" name="HICPP-5\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.2.4 Use std::call\_once rather than the Double-Checked Locking pattern" name="HICPP-18\_2\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.2.1 Ensure that pointer or array access is demonstrably within bounds of a valid object" name="HICPP-5\_2\_1" total="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="HIC++ 5.2.2 Ensure that functions do not call themselves, either directly or indirectly" name="HICPP-5\_2\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.4.2 Ensure that a constructor initializes explicitly all base classes and non-static data members" name="HICPP-12\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.4.3 Do not specify both an NSDMI and a member initializer in a constructor for the same non static member" name="HICPP-12\_4\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.4.1 Do not use the dynamic type of an object unless the object is fully constructed" name="HICPP-12\_4\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.4.1 Postpone variable definitions as long as possible" name="HICPP-6\_4\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 15.3.2 Ensure that a program does not result in a call to std::terminate" name="HICPP-15\_3\_2" total="3">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="HIC++ 16.1.1 Use the preprocessor only for implementing include guards, and including header files with include guards" name="HICPP-16\_1\_1" total="9">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Category>

<Category desc="HIC++ 16.1.2 Do not include a path specifier in filenames supplied in #include directives" name="HICPP-16\_1\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 11.2.1 Do not use friend declarations" name="HICPP-11\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 2.3.1 Do not use the C comment delimiters /\* ... \*/" name="HICPP-2\_3\_1" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 7.4.1 Ensure that any objects, functions or types to be used from a single translation unit are defined in an unnamed namespace in the main source file" name="HICPP-7\_4\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="HIC++ 7.4.3 Ensure that an object or a function used from multiple translation units is declared in a single header file" name="HICPP-7\_4\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.4.2 Ensure that an inline function, a function template, or a type used from multiple translation units is defined in a single header file" name="HICPP-7\_4\_2" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 16.1.5 Include directly the minimum number of headers required for compilation" name="HICPP-16\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 16.1.3 Match the filename in a #include directive to the one on the filesystem" name="HICPP-16\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 15.3.1 Do not access non-static members from a catch handler of constructor/destructor function try block" name="HICPP-15\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 16.1.4 Use &lt;> brackets for system and standard library headers. Use quotes for all other headers" name="HICPP-16\_1\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 2.3.2 Do not comment out code" name="HICPP-2\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 10.2.1 Use the override special identifier when overriding a virtual function" name="HICPP-10\_2\_1" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 17.3.4 Do not create smart pointers of array type" name="HICPP-17\_3\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 17.3.5 Do not create an rvalue reference of std::array" name="HICPP-17\_3\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 3.1.1 Do not hide declarations" name="HICPP-3\_1\_1" total="5">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 17.3.1 Do not use std::move on objects declared with const or const &amp; type" name="HICPP-17\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 14.1.1 Use variadic templates rather than an ellipsis" name="HICPP-14\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 17.3.2 Use std::forward to forward universal references" name="HICPP-17\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 17.3.3 Do not subsequently use the argument to std::forward" name="HICPP-17\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 3.5.1 Do not make any assumptions about the internal representation of a value or object" name="HICPP-3\_5\_1" total="4">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Category>

<Category desc="HIC++ 4.3.1 Do not convert an expression of wider floating point type to a narrower floating point type" name="HICPP-4\_3\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.1.10 Use static assert for assertions involving compile time constants" name="HICPP-7\_1\_10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 8.2.4 Do not pass std::unique ptr by const reference" name="HICPP-8\_2\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 8.2.3 Pass small objects with a trivial copy constructor by value" name="HICPP-8\_2\_3" total="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 8.2.2 Do not declare functions with an excessive number of parameters" name="HICPP-8\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 8.2.1 Make parameter names absent or identical in all declarations" name="HICPP-8\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.1.1 Do not use platform specific multi-threading facilities" name="HICPP-18\_1\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="HIC++ 5.1.6 Do not code side effects into the right-hand operands of: &amp;&amp;, ||, sizeof, typeid or a function passed to condition variable::wait" name="HICPP-5\_1\_6" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.1.5 Include a (possibly empty) parameter list in every lambda expression" name="HICPP-5\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.5.1 Ensure that the right hand operand of the division or remainder operators is demonstrably non-zero" name="HICPP-5\_5\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="HIC++ 5.1.2 Do not rely on the sequence of evaluation within an expression" name="HICPP-5\_1\_2" total="10">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.1.1 Use symbolic names instead of literal values in code" name="HICPP-5\_1\_1" total="1">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="HIC++ 5.1.4 Do not capture variables implicitly in a lambda" name="HICPP-5\_1\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.1.3 Use parentheses in expressions to specify the intent of the expression" name="HICPP-5\_1\_3" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.5.1 Define explicitly =default or =delete implicit special member functions of concrete classes" name="HICPP-12\_5\_1" total="1">

<Stats authTot="30;" authUrg="0;" total="30"/>

</Category>

<Category desc="HIC++ 12.5.2 Define special members =default if the behavior is equivalent" name="HICPP-12\_5\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.3.2 Ensure that execution of a function with a non-void return type ends in a return statement with a value" name="HICPP-6\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.1.1 Do not declare implicit user defined conversions" name="HICPP-12\_1\_1" total="2">

<Stats authTot="1;" authUrg="1;" total="1"/>

</Category>

<Category desc="HIC++ 1.2.1 Ensure that all statements are reachable" name="HICPP-1\_2\_1" total="10">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 1.2.2 Ensure that no expression or sub-expression is redundant" name="HICPP-1\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.3.1 Ensure that the label(s) for a jump statement or a switch condition appear later, in the same or an enclosing block" name="HICPP-6\_3\_1" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.1.4 Place CV-qualifiers on the right hand side of the type they apply to" name="HICPP-7\_1\_4" total="1">

<Stats authTot="4;" authUrg="0;" total="4"/>

</Category>

<Category desc="HIC++ 2.2.1 Do not use digraphs or trigraphs" name="HICPP-2\_2\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.1.3 Do not place type specifiers before non-type specifiers in a declaration" name="HICPP-7\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.1.6 Use class types or typedefs to abstract scalar quantities and standard integer types" name="HICPP-7\_1\_6" total="3">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Category>

<Category desc="HIC++ 7.1.5 Do not inline large functions" name="HICPP-7\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.5.1 Do not use the asm declaration" name="HICPP-7\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.1.8 Use auto id = expr when declaring a variable to have the same type as its initializer function call" name="HICPP-7\_1\_8" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 7.1.7 Use a trailing return type in preference to type disambiguation using typename" name="HICPP-7\_1\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.1.9 Do not explicitly specify the return type of a lambda" name="HICPP-7\_1\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 10.1.1 Ensure that access to base class subobjects does not require explicit disambiguation" name="HICPP-10\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 7.1.2 Use const whenever possible" name="HICPP-7\_1\_2" total="1">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Category>

<Category desc="HIC++ 7.1.1 Declare each identifier on a separate line in a separate declaration" name="HICPP-7\_1\_1" total="2">

<Stats authTot="14;" authUrg="0;" total="14"/>

</Category>

<Category desc="HIC++ 14.2.2 Do not explicitly specialize a function template that is overloaded with other templates" name="HICPP-14\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 17.2.1 Wrap use of the C Standard Library" name="HICPP-17\_2\_1" total="2">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 3.4.1 Do not return a reference or a pointer to an automatic variable defined within the function" name="HICPP-3\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 14.2.3 Declare extern an explicitly instantiated template" name="HICPP-14\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 3.4.3 Use RAII for resources" name="HICPP-3\_4\_3" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 14.2.1 Declare template specializations in the same file as the primary template they specialize." name="HICPP-14\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 3.4.2 Do not assign the address of a variable to a pointer with a greater lifetime" name="HICPP-3\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 4.2.2 Ensure that data loss does not demonstrably occur in an integral expression" name="HICPP-4\_2\_2" total="6">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 4.2.1 Ensure that the U suffix is applied to a literal used in a context requiring an unsigned integral expression" name="HICPP-4\_2\_1" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 8.3.4 Define =delete functions with parameters of type rvalue reference to const" name="HICPP-8\_3\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 8.3.3 Do not use default arguments" name="HICPP-8\_3\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 8.3.1 Do not write functions with an excessive McCabe Cyclomatic Complexity" name="HICPP-8\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 18.4.1 Do not use std::condition variable any on a std::mutex" name="HICPP-18\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.4.3 Do not convert from a base class to a derived class" name="HICPP-5\_4\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.4.2 Do not cast an expression to an enumeration type" name="HICPP-5\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.8.1 Do not use the conditional operator (?:) as a sub-expression" name="HICPP-5\_8\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 5.4.1 Only use casting forms: static cast (excl. void\*), dynamic cast or explicit constructor call" name="HICPP-5\_4\_1" total="3">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 9.1.5 Do not introduce virtual functions in a final class" name="HICPP-9\_1\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 17.1.1 Do not use std::vector&lt;bool>" name="HICPP-17\_1\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 9.1.3 Do not return non-const handles to class data from const member functions" name="HICPP-9\_1\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 9.1.4 Do not write member functions which return non-const handles to data less accessible than the member function" name="HICPP-9\_1\_4" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="HIC++ 9.1.1 Declare static any member function that does not require this. Alternatively, declare const any member function that does not modify the externally visible state of the object" name="HICPP-9\_1\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="HIC++ 9.1.2 Make default arguments the same or absent when overriding a virtual function" name="HICPP-9\_1\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 1.3.4 Do not use deprecated STL library features" name="HICPP-1\_3\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 1.3.3 Do not use the C Standard Library .h headers" name="HICPP-1\_3\_3" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="HIC++ 6.2.4 Only modify a for loop counter in the for expression" name="HICPP-6\_2\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 1.3.5 Do not use throw exception specifications" name="HICPP-1\_3\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.2.3 Do not alter a control or counter variable more than once in a loop" name="HICPP-6\_2\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 12.2.1 Declare virtual, private or protected the destructor of a type used as a base class" name="HICPP-12\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 1.3.2 Do not use the register keyword" name="HICPP-1\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 1.3.1 Do not use the increment operator (++) on a variable of type bool" name="HICPP-1\_3\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.2.2 Ensure that a loop has a single loop counter, an optional control variable, and is not degenerate" name="HICPP-6\_2\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="HIC++ 6.2.1 Implement a loop that only uses element values as a range-based loop" name="HICPP-6\_2\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Initialization" name="INIT" total="20">

<Stats authTot="12;" authUrg="0;" total="12"/>

</Category>

<Category desc="Joint Strike Fighter" name="JSF" total="278">

<Stats authTot="512;" authUrg="0;" total="512"/>

</Category>

<Category desc="Metrics" name="METRICS" total="43">

<Stats authTot="70;" authUrg="0;" total="70"/>

</Category>

<Category desc="MISRA C 1998" name="MISRA" total="60">

<Stats authTot="59;" authUrg="0;" total="59"/>

</Category>

<Category desc="MISRA C 2004" name="MISRA2004" total="221">

<Stats authTot="91;" authUrg="0;" total="91"/>

</Category>

<Category desc="MISRA C++ 2008" name="MISRA2008" total="336">

<Stats authTot="242;" authUrg="0;" total="242"/>

</Category>

<Category desc="MISRA C 2012 (Legacy)" name="MISRA2012" total="322">

<Category desc="MISRA C 2012 Rules" name="MISRA2012-RULE" total="279">

<Stats authTot="58;" authUrg="0;" total="58"/>

</Category>

<Category desc="MISRA C 2012 Directives" name="MISRA2012-DIR" total="43">

<Stats authTot="39;" authUrg="0;" total="39"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="MISRA C 2012" name="MISRAC2012" total="322">

<Category desc="Rule 11.3 (Required) A cast shall not be performed between a pointer to object type and a pointer to a different object type" name="MISRAC2012-RULE\_11\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.13 (Required) A line whose first token is # shall be a valid preprocessing directive" name="MISRAC2012-RULE\_20\_13" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.5 (Required) An external object or function shall be declared once in one and only one file" name="MISRAC2012-RULE\_8\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 11.4 (Advisory) A conversion should not be performed between a pointer to object and an integer type" name="MISRAC2012-RULE\_11\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.12 (Required) A macro parameter used as an operand to the # or ## operators, which is itself subject to further macro replacement, shall only be used as an operand to these operators" name="MISRAC2012-RULE\_20\_12" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.6 (Required) An identifier with external linkage shall have exactly one external definition" name="MISRAC2012-RULE\_8\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 11.5 (Advisory) A conversion should not be performed from pointer to void into pointer to object" name="MISRAC2012-RULE\_11\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 15.1 (Advisory) The goto statement should not be used" name="MISRAC2012-RULE\_15\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.11 (Required) A macro parameter immediately following a # operator shall not immediately be followed by a ## operator" name="MISRAC2012-RULE\_20\_11" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.3 (Required) All declarations of an object or function shall use the same names and type qualifiers" name="MISRAC2012-RULE\_8\_3" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 11.6 (Required) A cast shall not be performed between pointer to void and an arithmetic type" name="MISRAC2012-RULE\_11\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 15.2 (Required) The goto statement shall jump to a label declared later in the same function" name="MISRAC2012-RULE\_15\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.10 (Advisory) The # and ## preprocessor operators should not be used" name="MISRAC2012-RULE\_20\_10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.4 (Required) A compatible declaration shall be visible when an object or function with external linkage is defined" name="MISRAC2012-RULE\_8\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 11.7 (Required) A cast shall not be performed between pointer to object and a non-integer arithmetic type" name="MISRAC2012-RULE\_11\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 15.3 (Required) Any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement" name="MISRAC2012-RULE\_15\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.9 (Advisory) An object should be defined at block scope if its identifier only appears in a single function" name="MISRAC2012-RULE\_8\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 11.8 (Required) A cast shall not remove any const or volatile qualification from the type pointed to by a pointer" name="MISRAC2012-RULE\_11\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 15.4 (Advisory) There should be no more than one break or goto statement used to terminate any iteration statement" name="MISRAC2012-RULE\_15\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 11.9 (Required) The macro NULL shall be the only permitted form of integer null pointer constant" name="MISRAC2012-RULE\_11\_9" total="2">

<Stats authTot="22;" authUrg="0;" total="22"/>

</Category>

<Category desc="Rule 15.5 (Advisory) A function should have a single point of exit at the end" name="MISRAC2012-RULE\_15\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 19.1 (Mandatory) An object shall not be assigned or copied to an overlapping object" name="MISRAC2012-RULE\_19\_1" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.7 (Advisory) Functions and objects should not be defined with external linkage if they are referenced in only one translation unit" name="MISRAC2012-RULE\_8\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 15.6 (Required) The body of an iteration-statement or a selection-statement shall be a compound-statement" name="MISRAC2012-RULE\_15\_6" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 19.2 (Advisory) The union keyword should not be used" name="MISRAC2012-RULE\_19\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.8 (Required) The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage" name="MISRAC2012-RULE\_8\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.14 (Required) The validity of values received from external sources shall be checked" name="MISRAC2012-DIR\_4\_14" total="12">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 15.7 (Required) All if ... else if constructs shall be terminated with an else statement" name="MISRAC2012-RULE\_15\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.13 (Advisory) Functions which are designed to provide operations on a resource should be called in an appropriate sequence" name="MISRAC2012-DIR\_4\_13" total="6">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="Dir 4.12 (Required) Dynamic memory allocation shall not be used" name="MISRAC2012-DIR\_4\_12" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="Dir 4.11 (Required) The validity of values passed to library functions shall be checked" name="MISRAC2012-DIR\_4\_11" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.10 (Required) Precautions shall be taken in order to prevent the contents of a header file being included more than once" name="MISRAC2012-DIR\_4\_10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.14 (Required) All #else, #elif and #endif preprocessor directives shall reside in the same file as the #if, #ifdef or #ifndef directive to which they are related" name="MISRAC2012-RULE\_20\_14" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 4.1 (Required) Octal and hexadecimal escape sequences shall be terminated" name="MISRAC2012-RULE\_4\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 4.2 (Advisory) Trigraphs should not be used" name="MISRAC2012-RULE\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.1 (Required) Types shall be explicitly specified" name="MISRAC2012-RULE\_8\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.2 (Required) Function types shall be in prototype form with named parameters" name="MISRAC2012-RULE\_8\_2" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 11.1 (Required) Conversions shall not be performed between a pointer to a function and any other type" name="MISRAC2012-RULE\_11\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 11.2 (Required) Conversions shall not be performed between a pointer to an incomplete type and any other type" name="MISRAC2012-RULE\_11\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.3 (Required) The same file shall not be open for read and write access at the same time on different streams" name="MISRAC2012-RULE\_22\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.4 (Mandatory) There shall be no attempt to write to a stream which has been opened as read-only" name="MISRAC2012-RULE\_22\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.5 (Mandatory) A pointer to a FILE object shall not be dereferenced" name="MISRAC2012-RULE\_22\_5" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.6 (Mandatory) The value of a pointer to a FILE shall not be used after the associated stream has been closed" name="MISRAC2012-RULE\_22\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.7 (Required) The macro EOF shall only be compared with the unmodified return value from any Standard Library function capable of returning EOF" name="MISRAC2012-RULE\_22\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.8 (Required) The value of errno shall be set to zero prior to a call to an errno-setting-function" name="MISRAC2012-RULE\_22\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.9 (Required) The value of errno shall be tested against zero after calling an errno-setting-function" name="MISRAC2012-RULE\_22\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.1 (Required) All resources obtained dynamically by means of Standard Library functions shall be explicitly released" name="MISRAC2012-RULE\_22\_1" total="1">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="Rule 22.2 (Mandatory) A block of memory shall only be freed if it was allocated by means of a" name="MISRAC2012-RULE\_22\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 10.4 (Required) Both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category" name="MISRAC2012-RULE\_10\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 10.5 (Advisory) The value of an expression should not be cast to an inappropriate essential type" name="MISRAC2012-RULE\_10\_5" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 14.1 (Required) A loop counter shall not have essentially floating type" name="MISRAC2012-RULE\_14\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 10.6 (Required) The value of a composite expression shall not be assigned to an object with wider essential type" name="MISRAC2012-RULE\_10\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 14.2 (Required) A for loop shall be well-formed" name="MISRAC2012-RULE\_14\_2" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 7.4 (Required) A string literal shall not be assigned to an object unless the object's type is &quot;pointer to const-qualified char&quot;" name="MISRAC2012-RULE\_7\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 10.7 (Required) If a composite expression is used as one operand of an operator in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type" name="MISRAC2012-RULE\_10\_7" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 14.3 (Required) Controlling expressions shall not be invariant" name="MISRAC2012-RULE\_14\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 10.8 (Required) The value of a composite expression shall not be cast to a different essential type category or a wider essential type" name="MISRAC2012-RULE\_10\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 14.4 (Required) The controlling expression of an if statement and the controlling expression of an iteration-statement shall have essentially Boolean type" name="MISRAC2012-RULE\_14\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 18.1 (Required) A pointer resulting from arithmetic on a pointer operand shall address an element of the same array as that pointer operand" name="MISRAC2012-RULE\_18\_1" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 18.2 (Required) Subtraction between pointers shall only be applied to pointers that address elements of the same array" name="MISRAC2012-RULE\_18\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 18.3 (Required) The relational operators >, >=, &lt; and &lt;= shall not be applied to objects of pointer type except where they point into the same object" name="MISRAC2012-RULE\_18\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 18.4 (Advisory) The +, -, += and -= operators should not be applied to an expression of pointer type" name="MISRAC2012-RULE\_18\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 18.5 (Advisory) Declarations should contain no more than two levels of pointer nesting" name="MISRAC2012-RULE\_18\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 18.6 (Required) The address of an object with automatic storage shall not be copied to another object that persists after the first object has ceased to exist" name="MISRAC2012-RULE\_18\_6" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 18.7 (Required) Flexible array members shall not be declared" name="MISRAC2012-RULE\_18\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 18.8 (Required) Variable-length array types shall not be used" name="MISRAC2012-RULE\_18\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 3.2 (Required) Line-splicing shall not be used in // comments" name="MISRAC2012-RULE\_3\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 3.1 (Required) The character sequences /\* and // shall not be used within a comment" name="MISRAC2012-RULE\_3\_1" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 7.2 (Required) A &quot;u&quot; or &quot;U&quot; suffix shall be applied to all integer constants that are represented in an unsigned type" name="MISRAC2012-RULE\_7\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 10.1 (Required) Operands shall not be of an inappropriate essential type" name="MISRAC2012-RULE\_10\_1" total="7">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 7.3 (Required) The lowercase character &quot;l&quot; shall not be used in a literal suffix" name="MISRAC2012-RULE\_7\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 10.2 (Required) Expressions of essentially character type shall not be used inappropriately in addition and subtraction operations" name="MISRAC2012-RULE\_10\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 10.3 (Required) The value of an expression shall not be assigned to an object with a narrower essential type or of a different essential type category" name="MISRAC2012-RULE\_10\_3" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 7.1 (Required) Octal constants shall not be used" name="MISRAC2012-RULE\_7\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.14 (Required) The Standard Library function memcmp shall not be used to compare null terminated strings" name="MISRAC2012-RULE\_21\_14" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.4 (Required) The standard header file &lt;setjmp.h> shall not be used" name="MISRAC2012-RULE\_21\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.13 (Mandatory) Any value passed to a function in &lt;ctype.h> shall be representable as an unsigned char or be the value EOF" name="MISRAC2012-RULE\_21\_13" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.5 (Required) The standard header file &lt;signal.h> shall not be used" name="MISRAC2012-RULE\_21\_5" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.12 (Advisory) The exception handling features of &lt;fenv.h> should not be used" name="MISRAC2012-RULE\_21\_12" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.6 (Required) The Standard Library input/output functions shall not be used" name="MISRAC2012-RULE\_21\_6" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="Rule 21.11 (Required) The standard header file &lt;tgmath.h> shall not be used" name="MISRAC2012-RULE\_21\_11" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.7 (Required) The atof, atoi, atol and atoll functions of &lt;stdlib.h> shall not be used" name="MISRAC2012-RULE\_21\_7" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="Rule 21.10 (Required) The Standard Library time and date functions shall not be used" name="MISRAC2012-RULE\_21\_10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.8 (Required) The Standard Library termination functions of &lt;stdlib.h> shall not be used" name="MISRAC2012-RULE\_21\_8" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.9 (Required) The library functions bsearch and qsort of &lt;stdlib.h> shall not be used" name="MISRAC2012-RULE\_21\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.19 (Mandatory) The pointers returned by the Standard Library functions localeconv, getenv, setlocale or, strerror shall only be used as if they have pointer to const-qualified type" name="MISRAC2012-RULE\_21\_19" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.18 (Mandatory) The size\_t argument passed to any function in &lt;string.h> shall have an appropriate value" name="MISRAC2012-RULE\_21\_18" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.17 (Mandatory) Use of the string handling functions from &lt;string.h> shall not result in accesses beyond the bounds of the objects referenced by their pointer parameters" name="MISRAC2012-RULE\_21\_17" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.16 (Required) The pointer arguments to the Standard Library function memcmp shall point to either a pointer type, an essentially signed type, an essentially unsigned type, an essentially Boolean type or an essentially enum type" name="MISRAC2012-RULE\_21\_16" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.15 (Required) The pointer arguments to the Standard Library functions memcpy, memmove and memcmp shall be pointers to qualified or unqualified versions of compatible types" name="MISRAC2012-RULE\_21\_15" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.1 (Required) #define and #undef shall not be used on a reserved identifier or reserved macro name" name="MISRAC2012-RULE\_21\_1" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.2 (Required) A reserved identifier or macro name shall not be declared" name="MISRAC2012-RULE\_21\_2" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.3 (Required) The memory allocation and deallocation functions of &lt;stdlib.h> shall not be used" name="MISRAC2012-RULE\_21\_3" total="1">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="Rule 13.1 (Required) Initializer lists shall not contain persistent side effects" name="MISRAC2012-RULE\_13\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 13.2 (Required) The value of an expression and its persistent side effects shall be the same under all permitted evaluation orders" name="MISRAC2012-RULE\_13\_2" total="7">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 13.3 (Advisory) A full expression containing an increment (++) or decrement (--) operator should have no other potential side effects other than that caused by the increment or decrement operator" name="MISRAC2012-RULE\_13\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 13.4 (Advisory) The result of an assignment operator should not be used" name="MISRAC2012-RULE\_13\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 13.5 (Required) The right hand operand of a logical &amp;&amp; or || operator shall not contain persistent side effects" name="MISRAC2012-RULE\_13\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 17.1 (Required) The features of &lt;stdarg.h> shall not be used" name="MISRAC2012-RULE\_17\_1" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.21 (Required) The Standard Library function system of &lt;stdlib.h> shall not be used" name="MISRAC2012-RULE\_21\_21" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 13.6 (Required) The operand of the sizeof operator shall not contain any expression which has potential side effects" name="MISRAC2012-RULE\_13\_6" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 17.2 (Required) Functions shall not call themselves, either directly or indirectly" name="MISRAC2012-RULE\_17\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 21.20 (Mandatory) The pointer returned by the Standard Library functions asctime, ctime, gmtime, localtime, localeconv, getenv, setlocale or strerror shall not be used following a subsequent call to the same function" name="MISRAC2012-RULE\_21\_20" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 17.3 (Mandatory) A function shall not be declared implicitly" name="MISRAC2012-RULE\_17\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 17.4 (Mandatory) All exit paths from a function with non-void return type shall have an explicit return statement with an expression" name="MISRAC2012-RULE\_17\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 17.5 (Required) The function argument corresponding to a parameter declared to have an array type shall have an appropriate number of elements" name="MISRAC2012-RULE\_17\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 17.6 (Mandatory) The declaration of an array parameter shall not contain the static keyword between the [ ]" name="MISRAC2012-RULE\_17\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 17.7 (Required) The value returned by a function having non-void return type shall be used" name="MISRAC2012-RULE\_17\_7" total="2">

<Stats authTot="17;" authUrg="0;" total="17"/>

</Category>

<Category desc="Rule 17.8 (Advisory) A function parameter should not be modified" name="MISRAC2012-RULE\_17\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 2.3 (Advisory) A project should not contain unused type declarations" name="MISRAC2012-RULE\_2\_3" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 2.4 (Advisory) A project should not contain unused tag declarations" name="MISRAC2012-RULE\_2\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 2.1 (Required) A project shall not contain unreachable code" name="MISRAC2012-RULE\_2\_1" total="7">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 2.2 (Required) There shall be no dead code" name="MISRAC2012-RULE\_2\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 2.7 (Advisory) There should be no unused parameters in functions" name="MISRAC2012-RULE\_2\_7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 2.5 (Advisory) A project should not contain unused macro definitions" name="MISRAC2012-RULE\_2\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 6.1 (Required) Bit-fields shall only be declared with an appropriate type" name="MISRAC2012-RULE\_6\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 2.6 (Advisory) A function should not contain unused label declarations" name="MISRAC2012-RULE\_2\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 6.2 (Required) Single-bit named bit fields shall not be of a signed type" name="MISRAC2012-RULE\_6\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.5 (Advisory) #undef should not be used" name="MISRAC2012-RULE\_20\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.6 (Required) Tokens that look like a preprocessing directive shall not occur within a macro argument" name="MISRAC2012-RULE\_20\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.7 (Required) Expressions resulting from the expansion of macro parameters shall be enclosed in parentheses" name="MISRAC2012-RULE\_20\_7" total="1">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Category>

<Category desc="Rule 20.8 (Required) The controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1" name="MISRAC2012-RULE\_20\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.9 (Required) All identifiers used in the controlling expression of #if or #elif preprocessing directives shall be #define'd before evaluation" name="MISRAC2012-RULE\_20\_9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.1 (Advisory) #include directives should only be preceded by preprocessor directives or comments" name="MISRAC2012-RULE\_20\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.2 (Required) The ', &quot; or \ characters and the /\* or // character sequences shall not occur in a header file name" name="MISRAC2012-RULE\_20\_2" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.3 (Required) The #include directive shall be followed by either a &lt;filename> or &quot;filename&quot; sequence" name="MISRAC2012-RULE\_20\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 20.4 (Required) A macro shall not be defined with the same name as a keyword" name="MISRAC2012-RULE\_20\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 12.2 (Required) The right hand operand of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left hand operand" name="MISRAC2012-RULE\_12\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 5.8 (Required) Identifiers that define objects or functions with external linkage shall be unique" name="MISRAC2012-RULE\_5\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 9.4 (Required) An element of an object shall not be initialized more than once" name="MISRAC2012-RULE\_9\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 12.3 (Advisory) The comma operator should not be used" name="MISRAC2012-RULE\_12\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 5.9 (Advisory) Identifiers that define objects or functions with internal linkage should be unique" name="MISRAC2012-RULE\_5\_9" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 9.5 (Required) Where designated initializers are used to initialize an array object the size of the array shall be specified explicitly" name="MISRAC2012-RULE\_9\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 12.4 (Advisory) Evaluation of constant expressions should not lead to unsigned integer wrap-around" name="MISRAC2012-RULE\_12\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 5.6 (Required) A typedef name shall be a unique identifier" name="MISRAC2012-RULE\_5\_6" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 9.2 (Required) The initializer for an aggregate or union shall be enclosed in braces" name="MISRAC2012-RULE\_9\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 12.5 (Mandatory) The sizeof operator shall not have an operand which is a function parameter declared as &quot;array of type&quot;" name="MISRAC2012-RULE\_12\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 16.1 (Required) All switch statements shall be well-formed" name="MISRAC2012-RULE\_16\_1" total="8">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 5.7 (Required) A tag name shall be a unique identifier" name="MISRAC2012-RULE\_5\_7" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 9.3 (Required) Arrays shall not be partially initialized" name="MISRAC2012-RULE\_9\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 16.2 (Required) A switch label shall only be used when the most closely-enclosing compound statement is the body of a switch statement" name="MISRAC2012-RULE\_16\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 16.3 (Required) An unconditional break statement shall terminate every switch-clause" name="MISRAC2012-RULE\_16\_3" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 22.10 (Required) The value of errno shall only be tested when the last function to be called was an errno-setting-function" name="MISRAC2012-RULE\_22\_10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 16.4 (Required) Every switch statement shall have a default label" name="MISRAC2012-RULE\_16\_4" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 16.5 (Required) A default label shall appear as either the first or the last switch label of a switch statement" name="MISRAC2012-RULE\_16\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.9 (Advisory) A function should be used in preference to a function-like macro where they are interchangeable" name="MISRAC2012-DIR\_4\_9" total="1">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Category>

<Category desc="Rule 16.6 (Required) Every switch statement shall have at least two switch-clauses" name="MISRAC2012-RULE\_16\_6" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.8 (Advisory) If a pointer to a structure or union is never dereferenced within a translation unit, then the implementation of the object should be hidden" name="MISRAC2012-DIR\_4\_8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 16.7 (Required) A switch-expression shall not have essentially Boolean type" name="MISRAC2012-RULE\_16\_7" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.5 (Advisory) Identifiers in the same name space with overlapping visibility should be typographically unambiguous" name="MISRAC2012-DIR\_4\_5" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.4 (Advisory) Sections of code should not be &quot;commented out&quot;" name="MISRAC2012-DIR\_4\_4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.7 (Required) If a function returns error information, then that error information shall be tested" name="MISRAC2012-DIR\_4\_7" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.6 (Advisory) typedefs that indicate size and signedness should be used in place of the basic numerical types" name="MISRAC2012-DIR\_4\_6" total="3">

<Stats authTot="25;" authUrg="0;" total="25"/>

</Category>

<Category desc="Dir 4.1 (Required) Run-time failures shall be minimized" name="MISRAC2012-DIR\_4\_1" total="11">

<Stats authTot="2;" authUrg="0;" total="2"/>

</Category>

<Category desc="Rule 8.12 (Required) Within an enumerator list, the value of an implicitly-specified enumeration constant shall be unique" name="MISRAC2012-RULE\_8\_12" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.13 (Advisory) A pointer should point to a const-qualified type whenever possible" name="MISRAC2012-RULE\_8\_13" total="2">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="Dir 4.3 (Required) Assembly language shall be encapsulated and isolated" name="MISRAC2012-DIR\_4\_3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.14 (Required) The restrict type qualifier shall not be used" name="MISRAC2012-RULE\_8\_14" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Dir 4.2 (Advisory) All usage of assembly language should be documented" name="MISRAC2012-DIR\_4\_2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 1.1 (Required) The program shall contain no violations of the standard C syntax and constraints, and shall not exceed the implementation's translation limits" name="MISRAC2012-RULE\_1\_1" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.10 (Required) An inline function shall be declared with the static storage class" name="MISRAC2012-RULE\_8\_10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 8.11 (Advisory) When an array with external linkage is declared, its size should be explicitly specified" name="MISRAC2012-RULE\_8\_11" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 1.4 (Required) Emergent language features shall not be used" name="MISRAC2012-RULE\_1\_4" total="14">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 5.1 (Required) External identifiers shall be distinct" name="MISRAC2012-RULE\_5\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 1.3 (Required) There shall be no occurrence of undefined or critical unspecified behaviour" name="MISRAC2012-RULE\_1\_3" total="15">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="Rule 5.4 (Required) Macro identifiers shall be distinct" name="MISRAC2012-RULE\_5\_4" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 5.5 (Required) Identifiers shall be distinct from macro names" name="MISRAC2012-RULE\_5\_5" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 9.1 (Mandatory) The value of an object with automatic storage duration shall not be read before it has been set" name="MISRAC2012-RULE\_9\_1" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 5.2 (Required) Identifiers declared in the same scope and name space shall be distinct" name="MISRAC2012-RULE\_5\_2" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 12.1 (Advisory) The precedence of operators within expressions should be made explicit" name="MISRAC2012-RULE\_12\_1" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Rule 5.3 (Required) An identifier declared in an inner scope shall not hide an identifier declared in an outer scope" name="MISRAC2012-RULE\_5\_3" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Memory and Resource Management" name="MRM" total="60">

<Stats authTot="21;" authUrg="0;" total="21"/>

</Category>

<Category desc="Naming Conventions" name="NAMING" total="98">

<Category desc="Hungarian Notation" name="NAMING-HN" total="44">

<Stats authTot="53;" authUrg="0;" total="53"/>

</Category>

<Stats authTot="119;" authUrg="0;" total="119"/>

</Category>

<Category desc="Object Oriented" name="OOP" total="61">

<Stats authTot="9;" authUrg="1;" total="9"/>

</Category>

<Category desc="Optimization" name="OPT" total="43">

<Stats authTot="20;" authUrg="0;" total="20"/>

</Category>

<Category desc="OWASP Top 10 Most Critical Web Application Security Risks (2017)" name="OWASP2017" total="16">

<Category desc="OWASP A3 Sensitive Data Exposure" name="OWASP2017-A3" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A2 Broken Authentication" name="OWASP2017-A2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A1 Injection" name="OWASP2017-A1" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A10 Insufficient Logging&amp;Monitoring" name="OWASP2017-A10" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A6 Security Misconfiguration" name="OWASP2017-A6" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A5 Broken Access Control" name="OWASP2017-A5" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A4 XML External Entities (XXE)" name="OWASP2017-A4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP API Security Top 10 (2019)" name="OWASP2019" total="42">

<Category desc="OWASP API2 Broken User Authentication" name="OWASP2019-API2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP API4 Lack of Resources &amp; Rate Limiting" name="OWASP2019-API4" total="2">

<Stats authTot="2;" authUrg="1;" total="2"/>

</Category>

<Category desc="OWASP API3 Excessive Data Exposure" name="OWASP2019-API3" total="17">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP API10 Insufficient Logging &amp; Monitoring" name="OWASP2019-API10" total="2">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP API9 Improper Assets Management" name="OWASP2019-API9" total="7">

<Stats authTot="19;" authUrg="0;" total="19"/>

</Category>

<Category desc="OWASP API8 Injection" name="OWASP2019-API8" total="10">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP API7 Security Misconfiguration" name="OWASP2019-API7" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP Top 10 Most Critical Web Application Security Risks (2021)" name="OWASP2021" total="18">

<Category desc="OWASP A8 Software and Data Integrity Failures" name="OWASP2021-A8" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A7 Identification and Authentication Failures" name="OWASP2021-A7" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A5 Security Misconfiguration" name="OWASP2021-A5" total="4">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A9 Security Logging and Monitoring Failures" name="OWASP2021-A9" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A4 Insecure Design" name="OWASP2021-A4" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A3 Injection" name="OWASP2021-A3" total="6">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A2 Cryptographic Failures" name="OWASP2021-A2" total="1">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="OWASP A1 Broken Access Control" name="OWASP2021-A1" total="3">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Parser diagnostics" name="PARSER" total="2">

<Stats authTot="1;" authUrg="0;" total="1"/>

</Category>

<Category desc="Possible Bugs" name="PB" total="85">

<Stats authTot="9;" authUrg="0;" total="9"/>

</Category>

<Category desc="Physical File Organization" name="PFO" total="10">

<Stats authTot="8;" authUrg="0;" total="8"/>

</Category>

<Category desc="Portability" name="PORT" total="39">

<Stats authTot="15;" authUrg="0;" total="15"/>

</Category>

<Category desc="Preprocessor" name="PREPROC" total="29">

<Stats authTot="66;" authUrg="0;" total="66"/>

</Category>

<Category desc="Qt Best Practices" name="QT" total="18">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

<Category desc="Security" name="SECURITY" total="54">

<Stats authTot="12;" authUrg="0;" total="12"/>

</Category>

<Category desc="STL Best Practices" name="STL" total="42">

<Stats authTot="3;" authUrg="0;" total="3"/>

</Category>

<Category desc="Template" name="TEMPL" total="18">

<Stats authTot="0;" authUrg="0;" total="0"/>

</Category>

</CategoriesList>

<SeverityList>

<Severity id="1">

<Stats authTot="82;" authUrg="50;" total="82"/>

</Severity>

<Severity id="2">

<Stats authTot="801;" authUrg="0;" total="801"/>

</Severity>

<Severity id="3">

<Stats authTot="1466;" authUrg="0;" total="1466"/>

</Severity>

<Severity id="4">

<Stats authTot="473;" authUrg="0;" total="473"/>

</Severity>

<Severity id="5">

<Stats authTot="414;" authUrg="0;" total="414"/>

</Severity>

</SeverityList>

</Rules>

<StdViols>

<StdViol msg="Add comment containing the copyright information at the begin of file 'DeadLock.cpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-02" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'DeadLock.cpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'DeadLock.cpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-03" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'DeadLock.cpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Disallowed #include notation is being used: &quot;Shapes.hpp&quot;" ln="1" sev="2" auth="devtest" rule="PREPROC-09" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Disallowed #include notation is being used: &quot;Shapes.hpp&quot;" ln="1" sev="2" auth="devtest" rule="JSF-033" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implementation file 'DeadLock.cpp' should declare a local constant string that begins from characters &quot;@(#)&quot; " ln="1" sev="5" auth="devtest" rule="PFO-04" tool="c++test" cat="PFO" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implementation file 'DeadLock.cpp' should have the file name extension &quot;.cc&quot;" ln="1" sev="3" auth="devtest" rule="NAMING-38" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not a proper header file (\*.h ) is being included: &quot;Shapes.hpp&quot;" ln="1" sev="3" auth="devtest" rule="JSF-032" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not a proper header file (\*.h ) is being included: &quot;Shapes.hpp&quot;" ln="1" sev="3" auth="devtest" rule="PREPROC-08" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The assertion density is lower than two assertions per function" ln="1" sev="3" auth="devtest" rule="METRICS-31" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The filename 'DeadLock.cpp' should be in lowercase" ln="1" sev="3" auth="devtest" rule="NAMING-03" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#if pre-processor directive should not be used" ln="3" sev="3" auth="devtest" rule="PREPROC-10\_c" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#if pre-processor directive should not be used" ln="3" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_e" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#if pre-processor directive should not be used" ln="3" sev="3" auth="devtest" rule="HICPP-16\_1\_1-f" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #if preprocessor directive is used" ln="3" sev="3" auth="devtest" rule="PREPROC-11\_b" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #if preprocessor directive is used" ln="3" sev="2" auth="devtest" rule="JSF-026\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'THREAD'" ln="7" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'THREAD'" ln="7" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="7" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="7" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="7" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="7" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'THREAD' is enclosed in braces" ln="7" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'THREAD\_RETURN\_TYPE'" ln="8" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'THREAD\_RETURN\_TYPE'" ln="8" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="8" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="8" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="8" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="8" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'THREAD\_RETURN\_TYPE' is enclosed in braces" ln="8" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'THREAD\_CREATE' is defined without parentheses" ln="9" sev="3" auth="devtest" rule="MISRA-096" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'THREAD\_CREATE' is defined without parentheses" ln="9" sev="1" auth="devtest" rule="CERT\_C-PRE02-a" tool="c++test" cat="CERT\_C-PRE02" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="9" sev="2" auth="devtest" rule="MISRA2008-16\_0\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="9" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_9-a" tool="c++test" cat="MISRAC2012-DIR\_4\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="9" sev="3" auth="devtest" rule="MISRA2004-19\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="9" sev="2" auth="devtest" rule="JSF-029" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="9" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_9" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="9" sev="3" auth="devtest" rule="CERT\_C-PRE00-a" tool="c++test" cat="CERT\_C-PRE00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="9" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="9" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="9" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="9" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="9" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="9" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="9" sev="3" auth="devtest" rule="MISRA2004-19\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="64" locEndLn="9" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="9" sev="2" auth="devtest" rule="MISRA2008-16\_0\_6" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="64" locEndLn="9" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="9" sev="2" auth="devtest" rule="MISRA2012-RULE-20\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="64" locEndLn="9" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="9" sev="2" auth="devtest" rule="AUTOSAR-M16\_0\_6-a" tool="c++test" cat="AUTOSAR-M16\_0\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="64" locEndLn="9" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="9" sev="1" auth="devtest" rule="CERT\_C-PRE01-a" tool="c++test" cat="CERT\_C-PRE01" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="64" locEndLn="9" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="9" sev="2" auth="devtest" rule="MISRAC2012-RULE\_20\_7-a" tool="c++test" cat="MISRAC2012-RULE\_20\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="64" locEndLn="9" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="9" sev="3" auth="devtest" rule="MISRA2004-19\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="36" locEndLn="9" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="9" sev="2" auth="devtest" rule="MISRA2008-16\_0\_6" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="36" locEndLn="9" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="9" sev="2" auth="devtest" rule="MISRA2012-RULE-20\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="36" locEndLn="9" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="9" sev="2" auth="devtest" rule="AUTOSAR-M16\_0\_6-a" tool="c++test" cat="AUTOSAR-M16\_0\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="36" locEndLn="9" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="9" sev="1" auth="devtest" rule="CERT\_C-PRE01-a" tool="c++test" cat="CERT\_C-PRE01" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="36" locEndLn="9" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="9" sev="2" auth="devtest" rule="MISRAC2012-RULE\_20\_7-a" tool="c++test" cat="MISRAC2012-RULE\_20\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="36" locEndLn="9" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'THREAD\_CREATE' is enclosed in braces" ln="9" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'LOCK'" ln="11" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'LOCK'" ln="11" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="11" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="11" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="11" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="11" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="11" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'LOCK' is enclosed in braces" ln="11" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'LOCK\_ACQUIRE' is defined without parentheses" ln="12" sev="3" auth="devtest" rule="MISRA-096" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'LOCK\_ACQUIRE' is defined without parentheses" ln="12" sev="1" auth="devtest" rule="CERT\_C-PRE02-a" tool="c++test" cat="CERT\_C-PRE02" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="12" sev="2" auth="devtest" rule="MISRA2008-16\_0\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="12" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_9-a" tool="c++test" cat="MISRAC2012-DIR\_4\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="12" sev="3" auth="devtest" rule="MISRA2004-19\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="12" sev="2" auth="devtest" rule="JSF-029" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="12" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_9" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="12" sev="3" auth="devtest" rule="CERT\_C-PRE00-a" tool="c++test" cat="CERT\_C-PRE00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="12" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="12" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="12" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="12" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="12" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="12" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="12" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="12" sev="3" auth="devtest" rule="MISRA2004-19\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="12" sev="2" auth="devtest" rule="MISRA2008-16\_0\_6" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="12" sev="2" auth="devtest" rule="MISRA2012-RULE-20\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="12" sev="2" auth="devtest" rule="AUTOSAR-M16\_0\_6-a" tool="c++test" cat="AUTOSAR-M16\_0\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="12" sev="1" auth="devtest" rule="CERT\_C-PRE01-a" tool="c++test" cat="CERT\_C-PRE01" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="12" sev="2" auth="devtest" rule="MISRAC2012-RULE\_20\_7-a" tool="c++test" cat="MISRAC2012-RULE\_20\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'LOCK\_ACQUIRE' is enclosed in braces" ln="12" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'LOCK\_RELEASE' is defined without parentheses" ln="13" sev="3" auth="devtest" rule="MISRA-096" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'LOCK\_RELEASE' is defined without parentheses" ln="13" sev="1" auth="devtest" rule="CERT\_C-PRE02-a" tool="c++test" cat="CERT\_C-PRE02" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="13" sev="2" auth="devtest" rule="MISRA2008-16\_0\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="13" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_9-a" tool="c++test" cat="MISRAC2012-DIR\_4\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="13" sev="3" auth="devtest" rule="MISRA2004-19\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="13" sev="2" auth="devtest" rule="JSF-029" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="13" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_9" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="13" sev="3" auth="devtest" rule="CERT\_C-PRE00-a" tool="c++test" cat="CERT\_C-PRE00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="13" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="13" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="13" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="13" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="13" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="13" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="13" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="13" sev="3" auth="devtest" rule="MISRA2004-19\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="49" locEndLn="13" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="13" sev="2" auth="devtest" rule="MISRA2008-16\_0\_6" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="49" locEndLn="13" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="13" sev="2" auth="devtest" rule="MISRA2012-RULE-20\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="49" locEndLn="13" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="13" sev="2" auth="devtest" rule="AUTOSAR-M16\_0\_6-a" tool="c++test" cat="AUTOSAR-M16\_0\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="49" locEndLn="13" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="13" sev="1" auth="devtest" rule="CERT\_C-PRE01-a" tool="c++test" cat="CERT\_C-PRE01" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="49" locEndLn="13" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="13" sev="2" auth="devtest" rule="MISRAC2012-RULE\_20\_7-a" tool="c++test" cat="MISRAC2012-RULE\_20\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="49" locEndLn="13" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'LOCK\_RELEASE' is enclosed in braces" ln="13" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'SLEEP' is defined without parentheses" ln="15" sev="3" auth="devtest" rule="MISRA-096" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'SLEEP' is defined without parentheses" ln="15" sev="1" auth="devtest" rule="CERT\_C-PRE02-a" tool="c++test" cat="CERT\_C-PRE02" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="15" sev="2" auth="devtest" rule="MISRA2008-16\_0\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="15" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_9-a" tool="c++test" cat="MISRAC2012-DIR\_4\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="15" sev="3" auth="devtest" rule="MISRA2004-19\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="15" sev="2" auth="devtest" rule="JSF-029" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="15" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_9" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="15" sev="3" auth="devtest" rule="CERT\_C-PRE00-a" tool="c++test" cat="CERT\_C-PRE00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="15" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="15" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="15" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="15" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="15" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="15" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="15" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'SLEEP' is enclosed in braces" ln="15" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #else preprocessor directive is used" ln="17" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_c" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="17" locStartPos="0" locEndLn="17" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #else preprocessor directive is used" ln="17" sev="3" auth="devtest" rule="HICPP-16\_1\_1-c" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="17" locStartPos="0" locEndLn="17" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #else preprocessor directive is used" ln="17" sev="2" auth="devtest" rule="JSF-026" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="17" locStartPos="0" locEndLn="17" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #else preprocessor directive is used" ln="17" sev="3" auth="devtest" rule="PREPROC-11" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="17" locStartPos="0" locEndLn="17" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#if pre-processor directive should not be used" ln="19" sev="3" auth="devtest" rule="PREPROC-10\_c" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#if pre-processor directive should not be used" ln="19" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_e" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#if pre-processor directive should not be used" ln="19" sev="3" auth="devtest" rule="HICPP-16\_1\_1-f" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Line contains more than 120 characters: 302" ln="19" sev="3" auth="devtest" rule="JSF-041" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Line contains more than 120 characters: 302" ln="19" sev="3" auth="devtest" rule="METRICS-26" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Line has over 79 characters" ln="19" sev="3" auth="devtest" rule="FORMAT-04" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #if preprocessor directive is used" ln="19" sev="3" auth="devtest" rule="PREPROC-11\_b" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #if preprocessor directive is used" ln="19" sev="2" auth="devtest" rule="JSF-026\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use platform-specific multi-threading facilities: &lt;pthread.h>" ln="21" sev="3" auth="devtest" rule="HICPP-18\_1\_1-a" tool="c++test" cat="HICPP-18\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="21" locStartPos="0" locEndLn="21" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#else pre-processor directive should not be used" ln="24" sev="3" auth="devtest" rule="PREPROC-26" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="24" locStartPos="0" locEndLn="24" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#else pre-processor directive should not be used" ln="24" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-a" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="24" locStartPos="0" locEndLn="24" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #else preprocessor directive is used" ln="24" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_c" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="24" locStartPos="0" locEndLn="24" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #else preprocessor directive is used" ln="24" sev="3" auth="devtest" rule="HICPP-16\_1\_1-c" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="24" locStartPos="0" locEndLn="24" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #else preprocessor directive is used" ln="24" sev="2" auth="devtest" rule="JSF-026" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="24" locStartPos="0" locEndLn="24" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The #else preprocessor directive is used" ln="24" sev="3" auth="devtest" rule="PREPROC-11" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="24" locStartPos="0" locEndLn="24" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="30" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="30" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="30" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="30" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="30" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="30" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="30" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="30" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="30" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="30" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="30" locStartPos="0" locEndLn="30" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="35" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="35" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="35" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="35" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="35" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="35" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="35" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="35" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="35" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="35" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="35" locStartPos="0" locEndLn="35" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="37" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="37" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="37" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="37" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="37" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="37" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="37" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="37" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="37" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="37" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="37" locStartPos="0" locEndLn="37" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="39" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="39" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="39" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="39" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="39" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="39" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="39" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="39" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="39" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="39" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="39" locStartPos="11" locEndLn="39" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="41" sev="3" auth="devtest" rule="PREPROC-10\_b" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="41" locStartPos="0" locEndLn="41" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="41" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_d" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="41" locStartPos="0" locEndLn="41" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="41" sev="3" auth="devtest" rule="JSF-028\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="41" locStartPos="0" locEndLn="41" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="41" sev="3" auth="devtest" rule="HICPP-16\_1\_1-e" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="41" locStartPos="0" locEndLn="41" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="41" sev="3" auth="devtest" rule="PREPROC-26" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="41" locStartPos="0" locEndLn="41" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="41" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-a" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="41" locStartPos="0" locEndLn="41" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'THREAD'" ln="43" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="43" locStartPos="0" locEndLn="43" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'THREAD'" ln="43" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="43" locStartPos="0" locEndLn="43" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="43" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="43" locStartPos="0" locEndLn="43" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="43" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="43" locStartPos="0" locEndLn="43" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="43" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="43" locStartPos="0" locEndLn="43" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="43" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="43" locStartPos="0" locEndLn="43" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD'" ln="43" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="43" locStartPos="0" locEndLn="43" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'THREAD' is enclosed in braces" ln="43" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="43" locStartPos="0" locEndLn="43" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'THREAD\_RETURN\_TYPE' part of statement" ln="44" sev="4" auth="devtest" rule="PREPROC-04" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'THREAD\_RETURN\_TYPE'" ln="44" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'THREAD\_RETURN\_TYPE'" ln="44" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="44" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="44" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="44" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="44" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_RETURN\_TYPE'" ln="44" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'THREAD\_RETURN\_TYPE' is enclosed in braces" ln="44" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="44" locStartPos="0" locEndLn="44" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'THREAD\_CREATE' is defined without parentheses" ln="45" sev="3" auth="devtest" rule="MISRA-096" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'THREAD\_CREATE' is defined without parentheses" ln="45" sev="1" auth="devtest" rule="CERT\_C-PRE02-a" tool="c++test" cat="CERT\_C-PRE02" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="45" sev="2" auth="devtest" rule="MISRA2008-16\_0\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="45" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_9-a" tool="c++test" cat="MISRAC2012-DIR\_4\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="45" sev="3" auth="devtest" rule="MISRA2004-19\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="45" sev="2" auth="devtest" rule="JSF-029" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="45" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_9" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="45" sev="3" auth="devtest" rule="CERT\_C-PRE00-a" tool="c++test" cat="CERT\_C-PRE00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="45" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: THREAD\_CREATE" ln="45" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="45" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="45" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="45" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="45" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'THREAD\_CREATE'" ln="45" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="45" sev="3" auth="devtest" rule="MISRA2004-19\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="64" locEndLn="45" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="45" sev="2" auth="devtest" rule="MISRA2008-16\_0\_6" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="64" locEndLn="45" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="45" sev="2" auth="devtest" rule="MISRA2012-RULE-20\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="64" locEndLn="45" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="45" sev="2" auth="devtest" rule="AUTOSAR-M16\_0\_6-a" tool="c++test" cat="AUTOSAR-M16\_0\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="64" locEndLn="45" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="45" sev="1" auth="devtest" rule="CERT\_C-PRE01-a" tool="c++test" cat="CERT\_C-PRE01" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="45" locStartPos="64" locEndLn="45" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'func' should be enclosed in parentheses" ln="45" sev="2" auth="devtest" rule="MISRAC2012-RULE\_20\_7-a" tool="c++test" cat="MISRAC2012-RULE\_20\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="64" locEndLn="45" locEndPos="65" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="45" sev="3" auth="devtest" rule="MISRA2004-19\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="52" locEndLn="45" locEndPos="53" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="45" sev="2" auth="devtest" rule="MISRA2008-16\_0\_6" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="52" locEndLn="45" locEndPos="53" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="45" sev="2" auth="devtest" rule="MISRA2012-RULE-20\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="52" locEndLn="45" locEndPos="53" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="45" sev="2" auth="devtest" rule="AUTOSAR-M16\_0\_6-a" tool="c++test" cat="AUTOSAR-M16\_0\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="52" locEndLn="45" locEndPos="53" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="45" sev="1" auth="devtest" rule="CERT\_C-PRE01-a" tool="c++test" cat="CERT\_C-PRE01" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="52" locEndLn="45" locEndPos="53" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'thread' should be enclosed in parentheses" ln="45" sev="2" auth="devtest" rule="MISRAC2012-RULE\_20\_7-a" tool="c++test" cat="MISRAC2012-RULE\_20\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="52" locEndLn="45" locEndPos="53" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'THREAD\_CREATE' is enclosed in braces" ln="45" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="45" locStartPos="0" locEndLn="45" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'LOCK'" ln="47" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="47" locStartPos="0" locEndLn="47" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'LOCK'" ln="47" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="47" locStartPos="0" locEndLn="47" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="47" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="47" locStartPos="0" locEndLn="47" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="47" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="47" locStartPos="0" locEndLn="47" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="47" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="47" locStartPos="0" locEndLn="47" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="47" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="47" locStartPos="0" locEndLn="47" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK'" ln="47" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="47" locStartPos="0" locEndLn="47" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'LOCK' is enclosed in braces" ln="47" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="47" locStartPos="0" locEndLn="47" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'LOCK\_ACQUIRE' is defined without parentheses" ln="48" sev="3" auth="devtest" rule="MISRA-096" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'LOCK\_ACQUIRE' is defined without parentheses" ln="48" sev="1" auth="devtest" rule="CERT\_C-PRE02-a" tool="c++test" cat="CERT\_C-PRE02" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="48" sev="2" auth="devtest" rule="MISRA2008-16\_0\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="48" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_9-a" tool="c++test" cat="MISRAC2012-DIR\_4\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="48" sev="3" auth="devtest" rule="MISRA2004-19\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="48" sev="2" auth="devtest" rule="JSF-029" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="48" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_9" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="48" sev="3" auth="devtest" rule="CERT\_C-PRE00-a" tool="c++test" cat="CERT\_C-PRE00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="48" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_ACQUIRE" ln="48" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="48" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="48" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="48" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="48" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_ACQUIRE'" ln="48" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="48" sev="3" auth="devtest" rule="MISRA2004-19\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="47" locEndLn="48" locEndPos="48" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="48" sev="2" auth="devtest" rule="MISRA2008-16\_0\_6" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="47" locEndLn="48" locEndPos="48" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="48" sev="2" auth="devtest" rule="MISRA2012-RULE-20\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="47" locEndLn="48" locEndPos="48" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="48" sev="2" auth="devtest" rule="AUTOSAR-M16\_0\_6-a" tool="c++test" cat="AUTOSAR-M16\_0\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="47" locEndLn="48" locEndPos="48" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="48" sev="1" auth="devtest" rule="CERT\_C-PRE01-a" tool="c++test" cat="CERT\_C-PRE01" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="47" locEndLn="48" locEndPos="48" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="48" sev="2" auth="devtest" rule="MISRAC2012-RULE\_20\_7-a" tool="c++test" cat="MISRAC2012-RULE\_20\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="47" locEndLn="48" locEndPos="48" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'LOCK\_ACQUIRE' is enclosed in braces" ln="48" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="48" locStartPos="0" locEndLn="48" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'LOCK\_RELEASE' is defined without parentheses" ln="49" sev="3" auth="devtest" rule="MISRA-096" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'LOCK\_RELEASE' is defined without parentheses" ln="49" sev="1" auth="devtest" rule="CERT\_C-PRE02-a" tool="c++test" cat="CERT\_C-PRE02" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="49" sev="2" auth="devtest" rule="MISRA2008-16\_0\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="49" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_9-a" tool="c++test" cat="MISRAC2012-DIR\_4\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="49" sev="3" auth="devtest" rule="MISRA2004-19\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="49" sev="2" auth="devtest" rule="JSF-029" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="49" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_9" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="49" sev="3" auth="devtest" rule="CERT\_C-PRE00-a" tool="c++test" cat="CERT\_C-PRE00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="49" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: LOCK\_RELEASE" ln="49" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="49" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="49" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="49" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="49" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'LOCK\_RELEASE'" ln="49" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="49" sev="3" auth="devtest" rule="MISRA2004-19\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="49" locEndLn="49" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="49" sev="2" auth="devtest" rule="MISRA2008-16\_0\_6" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="49" locEndLn="49" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="49" sev="2" auth="devtest" rule="MISRA2012-RULE-20\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="49" locEndLn="49" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="49" sev="2" auth="devtest" rule="AUTOSAR-M16\_0\_6-a" tool="c++test" cat="AUTOSAR-M16\_0\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="49" locEndLn="49" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="49" sev="1" auth="devtest" rule="CERT\_C-PRE01-a" tool="c++test" cat="CERT\_C-PRE01" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="49" locStartPos="49" locEndLn="49" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Each instance of parameter: 'lock' should be enclosed in parentheses" ln="49" sev="2" auth="devtest" rule="MISRAC2012-RULE\_20\_7-a" tool="c++test" cat="MISRAC2012-RULE\_20\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="49" locEndLn="49" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'LOCK\_RELEASE' is enclosed in braces" ln="49" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="49" locStartPos="0" locEndLn="49" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'SLEEP' is defined without parentheses" ln="51" sev="3" auth="devtest" rule="MISRA-096" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Body of macro 'SLEEP' is defined without parentheses" ln="51" sev="1" auth="devtest" rule="CERT\_C-PRE02-a" tool="c++test" cat="CERT\_C-PRE02" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="51" sev="2" auth="devtest" rule="MISRA2008-16\_0\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="51" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_9-a" tool="c++test" cat="MISRAC2012-DIR\_4\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="51" sev="3" auth="devtest" rule="MISRA2004-19\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="51" sev="2" auth="devtest" rule="JSF-029" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="51" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_9" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="51" sev="3" auth="devtest" rule="CERT\_C-PRE00-a" tool="c++test" cat="CERT\_C-PRE00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="51" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not define function-like macro: SLEEP" ln="51" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="51" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="51" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="51" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="51" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'SLEEP'" ln="51" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Not all body of macro 'SLEEP' is enclosed in braces" ln="51" sev="3" auth="devtest" rule="PREPROC-14" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="51" locStartPos="0" locEndLn="51" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="53" sev="3" auth="devtest" rule="PREPROC-10\_b" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="53" locStartPos="0" locEndLn="53" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="53" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_d" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="53" locStartPos="0" locEndLn="53" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="53" sev="3" auth="devtest" rule="JSF-028\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="53" locStartPos="0" locEndLn="53" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="#endif pre-processor directive should not be used" ln="53" sev="3" auth="devtest" rule="HICPP-16\_1\_1-e" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="53" locStartPos="0" locEndLn="53" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Define constant 'MAX\_OBJECTS' using 'const' or 'enum' instead of '#define'" ln="55" sev="2" auth="devtest" rule="JSF-030" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Define constant 'MAX\_OBJECTS' using 'const' or 'enum' instead of '#define'" ln="55" sev="3" auth="devtest" rule="CODSTA-03" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'MAX\_OBJECTS'" ln="55" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'MAX\_OBJECTS'" ln="55" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'MAX\_OBJECTS'" ln="55" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'MAX\_OBJECTS'" ln="55" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'MAX\_OBJECTS'" ln="55" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'MAX\_OBJECTS'" ln="55" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'MAX\_OBJECTS'" ln="55" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Replace the 'MAX\_OBJECTS' macro with a constant variable" ln="55" sev="3" auth="devtest" rule="CODSTA-37" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="55" locStartPos="0" locEndLn="55" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Define constant 'STEP' using 'const' or 'enum' instead of '#define'" ln="56" sev="2" auth="devtest" rule="JSF-030" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Define constant 'STEP' using 'const' or 'enum' instead of '#define'" ln="56" sev="3" auth="devtest" rule="CODSTA-03" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'STEP'" ln="56" sev="2" auth="devtest" rule="MISRA2008-16\_2\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use #define to define constant: 'STEP'" ln="56" sev="3" auth="devtest" rule="PREPROC-12" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'STEP'" ln="56" sev="3" auth="devtest" rule="JSF-031" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'STEP'" ln="56" sev="2" auth="devtest" rule="MISRA2008-16\_2\_1\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'STEP'" ln="56" sev="3" auth="devtest" rule="HICPP-16\_1\_1-a" tool="c++test" cat="HICPP-16\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'STEP'" ln="56" sev="3" auth="devtest" rule="PREPROC-01" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not use the macro definition 'STEP'" ln="56" sev="2" auth="devtest" rule="AUTOSAR-A16\_0\_1-d" tool="c++test" cat="AUTOSAR-A16\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Replace the 'STEP' macro with a constant variable" ln="56" sev="3" auth="devtest" rule="CODSTA-37" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="56" locStartPos="0" locEndLn="56" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'condition' shall be declared as unsigned int or signed int" ln="58" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'condition' as const" ln="58" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'condition' as const" ln="58" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'condition' as const" ln="58" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'condition' as const" ln="58" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'condition' as const" ln="58" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'message' as const" ln="58" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'message' as const" ln="58" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'message' as const" ln="58" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'message' as const" ln="58" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'message' as const" ln="58" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has Cyclomatic Complexity value: 2" ln="58" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has Essential Complexity value: 1" ln="58" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has external linkage and is not declared in the header" ln="58" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has external linkage and is not declared in the header" ln="58" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has external linkage and is not declared in the header" ln="58" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has external linkage and is not declared in the header" ln="58" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has external linkage and is not declared in the header" ln="58" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has external linkage and is not declared in the header" ln="58" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'assertion' has external linkage and is not declared in the header" ln="58" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'assertion' is declared in global namespace" ln="58" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'assertion' is declared in global namespace" ln="58" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'assertion' is declared in global namespace" ln="58" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'assertion' is declared in global namespace" ln="58" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: assertion" ln="58" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Parameter 'message' is not validated before use" ln="58" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Parameter 'message' is not validated before use" ln="58" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Return type is not placed in line before function 'assertion'" ln="58" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'assertion' function should be preceded by a comment that contains the '@brief' tag" ln="58" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'assertion' function should be preceded by a comment that contains the '@brief' tag" ln="58" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'condition' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="58" sev="3" auth="devtest" rule="NAMING-HN-22" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'condition' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="58" sev="3" auth="devtest" rule="NAMING-HN-43" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'condition' identifier should have the 'n' prefix followed by a capital letter or an underscore" ln="58" sev="3" auth="devtest" rule="NAMING-HN-30" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="19" locEndLn="58" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'condition' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="58" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'condition' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="58" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'const' qualifier should be placed on the right hand side of the type" ln="58" sev="3" auth="devtest" rule="FORMAT-47\_a" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="30" locEndLn="58" locEndPos="31" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'const' qualifier should be placed on the right hand side of the type" ln="58" sev="3" auth="devtest" rule="HICPP-7\_1\_4-a" tool="c++test" cat="HICPP-7\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="30" locEndLn="58" locEndPos="31" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'message' identifier should have the 'c' prefix followed by a capital letter or an underscore" ln="58" sev="3" auth="devtest" rule="NAMING-HN-05" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'message' identifier should have the 'c' prefix followed by a capital letter or an underscore" ln="58" sev="3" auth="devtest" rule="NAMING-HN-09" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'message' identifier should have the 'c' prefix followed by a capital letter or an underscore" ln="58" sev="3" auth="devtest" rule="NAMING-HN-10" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'message' identifier should have the 'p' prefix" ln="58" sev="3" auth="devtest" rule="NAMING-HN-34" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'message' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="58" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'message' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="58" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'char' should not be used" ln="58" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="36" locEndLn="58" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'char' should not be used" ln="58" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="36" locEndLn="58" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="58" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="58" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="58" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="58" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="58" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="58" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="58" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="58" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'assertion' function is not preceded by a comment" ln="58" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'assertion' function is not preceded by a comment" ln="58" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="5" locEndLn="58" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared: message" ln="58" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer type is declared: message" ln="58" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="42" locEndLn="58" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'char' basic numerical type" ln="58" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="36" locEndLn="58" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'char' basic numerical type" ln="58" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="36" locEndLn="58" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="58" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="58" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="58" locStartPos="15" locEndLn="58" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="59" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="59" locStartPos="0" locEndLn="59" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="In 'assertion' function use positive logic rather than negative logic whenever possible" ln="60" sev="5" auth="devtest" rule="CODSTA-46" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="5" locEndLn="60" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="60" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="60" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="60" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="60" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="60" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="60" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="60" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="60" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="60" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="0" locEndLn="60" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'if' statement doesn't have an 'else' clause" ln="60" sev="3" auth="devtest" rule="CODSTA-23" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="1" locEndLn="60" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="60" sev="3" auth="devtest" rule="CODSTA-CPP-67" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="5" locEndLn="60" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="60" sev="2" auth="devtest" rule="AUTOSAR-M5\_3\_1-a" tool="c++test" cat="AUTOSAR-M5\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="5" locEndLn="60" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="60" sev="2" auth="devtest" rule="MISRA2008-5\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="5" locEndLn="60" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' is not 'effectively Boolean'" ln="60" sev="3" auth="devtest" rule="MISRA2004-12\_6\_a" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="60" locStartPos="5" locEndLn="60" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Exception of type 'const char \*' is thrown by pointer; throw by value instead" ln="61" sev="4" auth="devtest" rule="MISRA2008-15\_0\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="2" locEndLn="61" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Exception of type 'const char \*' is thrown by pointer; throw by value instead" ln="61" sev="2" auth="devtest" rule="AUTOSAR-A15\_1\_2-a" tool="c++test" cat="AUTOSAR-A15\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="2" locEndLn="61" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Exception of type 'const char \*' is thrown by pointer; throw by value instead" ln="61" sev="3" auth="devtest" rule="EXCEPT-09" tool="c++test" cat="EXCEPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="2" locEndLn="61" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="61" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="0" locEndLn="61" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="61" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="0" locEndLn="61" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="61" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="0" locEndLn="61" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="61" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="1" locEndLn="61" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="61" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="1" locEndLn="61" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="61" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="1" locEndLn="61" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Throw only instances of std::exception" ln="61" sev="3" auth="devtest" rule="HICPP-15\_1\_1-a" tool="c++test" cat="HICPP-15\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="2" locEndLn="61" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Throw only instances of std::exception" ln="61" sev="4" auth="devtest" rule="AUTOSAR-A15\_1\_1-a" tool="c++test" cat="AUTOSAR-A15\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="2" locEndLn="61" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="c++ exception handling structure is used in function 'assertion'" ln="61" sev="2" auth="devtest" rule="JSF-208" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="2" locEndLn="61" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="c++ exception handling structure is used in function 'assertion'" ln="61" sev="2" auth="devtest" rule="EXCEPT-05" tool="c++test" cat="EXCEPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="61" locStartPos="2" locEndLn="61" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="62" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="62" locStartPos="0" locEndLn="62" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="62" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="62" locStartPos="0" locEndLn="62" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="62" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="62" locStartPos="0" locEndLn="62" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'changePositionMutex' should be encapsulated in a class" ln="65" sev="5" auth="devtest" rule="CODSTA-CPP-18" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'changePositionMutex'" ln="65" sev="3" auth="devtest" rule="JSF-207" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'changePositionMutex'" ln="65" sev="3" auth="devtest" rule="CODSTA-CPP-82" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'changePositionMutex' is declared" ln="65" sev="5" auth="devtest" rule="MISRA-022" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'changePositionMutex' is declared in global namespace" ln="65" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'changePositionMutex' is declared in global namespace" ln="65" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'changePositionMutex' is declared in global namespace" ln="65" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'changePositionMutex' is declared in global namespace" ln="65" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: changePositionMutex" ln="65" sev="3" auth="devtest" rule="NAMING-18" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'changePositionMutex' uninitialized when declared" ln="65" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'changePositionMutex' identifier should have the 's' prefix" ln="65" sev="3" auth="devtest" rule="NAMING-HN-38" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'changePositionMutex' variable should be commented" ln="65" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'changePositionMutex' variable should be commented" ln="65" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'static' keyword is used in 'changePositionMutex' variable declaration" ln="65" sev="3" auth="devtest" rule="HICPP-7\_4\_1-a" tool="c++test" cat="HICPP-7\_4\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'static' keyword is used in 'changePositionMutex' variable declaration" ln="65" sev="3" auth="devtest" rule="CODSTA-CPP-50" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'changePositionMutex' should be composed only of lowercase letters" ln="65" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'changePositionMutex' should be composed only of lowercase letters" ln="65" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: changePositionMutex" ln="65" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: changePositionMutex" ln="65" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use of variable &quot;changePositionMutex&quot; with static storage duration is not allowed" ln="65" sev="3" auth="devtest" rule="HICPP-3\_3\_1-a" tool="c++test" cat="HICPP-3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="65" locStartPos="23" locEndLn="65" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'ring' should be encapsulated in a class" ln="67" sev="5" auth="devtest" rule="CODSTA-CPP-18" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Argument type conversion from &quot;shorter&quot; type to &quot;longer&quot; type" ln="67" sev="3" auth="devtest" rule="PORT-16" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'ring'" ln="67" sev="3" auth="devtest" rule="JSF-207" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'ring'" ln="67" sev="3" auth="devtest" rule="CODSTA-CPP-82" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' has external linkage and is not declared in the header" ln="67" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' has external linkage and is not declared in the header" ln="67" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' has external linkage and is not declared in the header" ln="67" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' has external linkage and is not declared in the header" ln="67" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' has external linkage and is not declared in the header" ln="67" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' has external linkage and is not declared in the header" ln="67" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' has external linkage and is not declared in the header" ln="67" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' is declared" ln="67" sev="5" auth="devtest" rule="MISRA-022" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' is declared in global namespace" ln="67" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' is declared in global namespace" ln="67" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' is declared in global namespace" ln="67" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'ring' is declared in global namespace" ln="67" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion from integral to floating type in function's argument" ln="67" sev="2" auth="devtest" rule="MISRA2008-5\_0\_5\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="24" locEndLn="67" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion from integral to floating type in function's argument" ln="67" sev="3" auth="devtest" rule="CERT\_C-EXP39-a" tool="c++test" cat="CERT\_C-EXP39" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="24" locEndLn="67" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion from integral to floating type in function's argument" ln="67" sev="2" auth="devtest" rule="AUTOSAR-M5\_0\_5-a" tool="c++test" cat="AUTOSAR-M5\_0\_5" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="24" locEndLn="67" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion from integral to floating type in function's argument" ln="67" sev="3" auth="devtest" rule="MISRA2004-10\_1\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="24" locEndLn="67" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: ring" ln="67" sev="3" auth="devtest" rule="NAMING-18" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Point' constructor called during the creation of the 'ring' object should be declared as exception-safe" ln="67" sev="2" auth="devtest" rule="AUTOSAR-M15\_3\_1-a" tool="c++test" cat="AUTOSAR-M15\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Point' constructor called during the creation of the 'ring' object should be declared as exception-safe" ln="67" sev="2" auth="devtest" rule="CERT\_CPP-ERR58-a" tool="c++test" cat="CERT\_CPP-ERR58" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Point' constructor called during the creation of the 'ring' object should be declared as exception-safe" ln="67" sev="2" auth="devtest" rule="MISRA2008-15\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Point' constructor called during the creation of the 'ring' object should be declared as exception-safe" ln="67" sev="2" auth="devtest" rule="AUTOSAR-A15\_5\_3-f" tool="c++test" cat="AUTOSAR-A15\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Point' constructor called during the creation of the 'ring' object should be declared as exception-safe" ln="67" sev="3" auth="devtest" rule="CERT\_CPP-ERR50-f" tool="c++test" cat="CERT\_CPP-ERR50" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Point' constructor called during the creation of the 'ring' object should be declared as exception-safe" ln="67" sev="3" auth="devtest" rule="EXCEPT-08" tool="c++test" cat="EXCEPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Point' constructor called during the creation of the 'ring' object should be declared as exception-safe" ln="67" sev="2" auth="devtest" rule="MISRA2008-15\_5\_3\_f" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'ring' variable should be commented" ln="67" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'ring' variable should be commented" ln="67" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The declaration of variable 'ring' contains a non-constant initializer" ln="67" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_2-a" tool="c++test" cat="AUTOSAR-A3\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The declaration of variable 'ring' contains a non-constant initializer" ln="67" sev="3" auth="devtest" rule="CODSTA-MCPP-32" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'ring' variable should contain a braced initializer" ln="67" sev="2" auth="devtest" rule="AUTOSAR-A8\_5\_2-a" tool="c++test" cat="AUTOSAR-A8\_5\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'ring' variable should contain a braced initializer" ln="67" sev="3" auth="devtest" rule="CODSTA-MCPP-38" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The type 'int' of function argument number '2' does not match declared type 'double'" ln="67" sev="3" auth="devtest" rule="PB-11" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use of variable &quot;ring&quot; with static storage duration is not allowed" ln="67" sev="3" auth="devtest" rule="HICPP-3\_3\_1-a" tool="c++test" cat="HICPP-3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Variable 'ring' should be declared using 'auto' specifier" ln="67" sev="3" auth="devtest" rule="HICPP-7\_1\_8-a" tool="c++test" cat="HICPP-7\_1\_8" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="67" locStartPos="7" locEndLn="67" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'participants' should be encapsulated in a class" ln="68" sev="5" auth="devtest" rule="CODSTA-CPP-18" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'participants'" ln="68" sev="3" auth="devtest" rule="JSF-207" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'participants'" ln="68" sev="3" auth="devtest" rule="CODSTA-CPP-82" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declaration of variable 'participants' contains more than one level of pointer indirection" ln="68" sev="3" auth="devtest" rule="HICPP-8\_1\_1-a" tool="c++test" cat="HICPP-8\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declaration of variable 'participants' contains more than one level of pointer indirection" ln="68" sev="3" auth="devtest" rule="CODSTA-89" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' has external linkage and is not declared in the header" ln="68" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' has external linkage and is not declared in the header" ln="68" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' has external linkage and is not declared in the header" ln="68" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' has external linkage and is not declared in the header" ln="68" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' has external linkage and is not declared in the header" ln="68" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' has external linkage and is not declared in the header" ln="68" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' has external linkage and is not declared in the header" ln="68" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' is declared" ln="68" sev="5" auth="devtest" rule="MISRA-022" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' is declared in global namespace" ln="68" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' is declared in global namespace" ln="68" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' is declared in global namespace" ln="68" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participants' is declared in global namespace" ln="68" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: participants" ln="68" sev="3" auth="devtest" rule="NAMING-18" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participants' array should not be used" ln="68" sev="3" auth="devtest" rule="STL-37" tool="c++test" cat="STL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participants' array should not be used" ln="68" sev="2" auth="devtest" rule="AUTOSAR-A18\_1\_1-a" tool="c++test" cat="AUTOSAR-A18\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participants' identifier should have the 'rg' prefix" ln="68" sev="3" auth="devtest" rule="NAMING-HN-01" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participants' variable should be commented" ln="68" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participants' variable should be commented" ln="68" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operator '\*', used for 'participants' declaration, should be directly connected with the type" ln="68" sev="3" auth="devtest" rule="JSF-062" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="6" locEndLn="68" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operator '\*', used for 'participants' declaration, should be directly connected with the type" ln="68" sev="4" auth="devtest" rule="FORMAT-32" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="6" locEndLn="68" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: participants" ln="68" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: participants" ln="68" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use of variable &quot;participants&quot; with static storage duration is not allowed" ln="68" sev="3" auth="devtest" rule="HICPP-3\_3\_1-a" tool="c++test" cat="HICPP-3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="68" locStartPos="7" locEndLn="68" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'participantsCount' shall be declared as unsigned int or signed int" ln="69" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'participantsCount' should be encapsulated in a class" ln="69" sev="5" auth="devtest" rule="CODSTA-CPP-18" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'participantsCount'" ln="69" sev="3" auth="devtest" rule="JSF-207" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'participantsCount'" ln="69" sev="3" auth="devtest" rule="CODSTA-CPP-82" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' has external linkage and is not declared in the header" ln="69" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' has external linkage and is not declared in the header" ln="69" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' has external linkage and is not declared in the header" ln="69" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' has external linkage and is not declared in the header" ln="69" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' has external linkage and is not declared in the header" ln="69" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' has external linkage and is not declared in the header" ln="69" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' has external linkage and is not declared in the header" ln="69" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' is declared" ln="69" sev="5" auth="devtest" rule="MISRA-022" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' is declared in global namespace" ln="69" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' is declared in global namespace" ln="69" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' is declared in global namespace" ln="69" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'participantsCount' is declared in global namespace" ln="69" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: participantsCount" ln="69" sev="3" auth="devtest" rule="NAMING-18" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: participantsCount" ln="69" sev="3" auth="devtest" rule="NAMING-19" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participantsCount' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="69" sev="3" auth="devtest" rule="NAMING-HN-22" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participantsCount' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="69" sev="3" auth="devtest" rule="NAMING-HN-43" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participantsCount' identifier should have the 'n' prefix followed by a capital letter or an underscore" ln="69" sev="3" auth="devtest" rule="NAMING-HN-30" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participantsCount' variable should be commented" ln="69" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'participantsCount' variable should be commented" ln="69" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="69" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="69" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="69" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="69" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="69" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="69" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="69" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="69" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'participantsCount' should be composed only of lowercase letters" ln="69" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'participantsCount' should be composed only of lowercase letters" ln="69" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable &quot;participantsCount&quot; was not initialized when declared" ln="69" sev="2" auth="devtest" rule="JSF-142\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable &quot;participantsCount&quot; was not initialized when declared" ln="69" sev="3" auth="devtest" rule="INIT-03" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use of variable &quot;participantsCount&quot; with static storage duration is not allowed" ln="69" sev="3" auth="devtest" rule="HICPP-3\_3\_1-a" tool="c++test" cat="HICPP-3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="4" locEndLn="69" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="69" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="69" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="69" locStartPos="0" locEndLn="69" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'exitGame' shall be declared as unsigned int or signed int" ln="70" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'exitGame' should be encapsulated in a class" ln="70" sev="5" auth="devtest" rule="CODSTA-CPP-18" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'exitGame'" ln="70" sev="3" auth="devtest" rule="JSF-207" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'exitGame'" ln="70" sev="3" auth="devtest" rule="CODSTA-CPP-82" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' has external linkage and is not declared in the header" ln="70" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' has external linkage and is not declared in the header" ln="70" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' has external linkage and is not declared in the header" ln="70" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' has external linkage and is not declared in the header" ln="70" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' has external linkage and is not declared in the header" ln="70" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' has external linkage and is not declared in the header" ln="70" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' has external linkage and is not declared in the header" ln="70" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' is declared" ln="70" sev="5" auth="devtest" rule="MISRA-022" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' is declared in global namespace" ln="70" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' is declared in global namespace" ln="70" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' is declared in global namespace" ln="70" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' is declared in global namespace" ln="70" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: exitGame" ln="70" sev="3" auth="devtest" rule="NAMING-18" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: exitGame" ln="70" sev="3" auth="devtest" rule="NAMING-19" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'exitGame' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="70" sev="3" auth="devtest" rule="NAMING-HN-22" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'exitGame' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="70" sev="3" auth="devtest" rule="NAMING-HN-43" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'exitGame' identifier should have the 'n' prefix followed by a capital letter or an underscore" ln="70" sev="3" auth="devtest" rule="NAMING-HN-30" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'exitGame' variable should be commented" ln="70" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'exitGame' variable should be commented" ln="70" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="70" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="70" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="70" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="70" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="70" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="70" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="70" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="70" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'exitGame' should be composed only of lowercase letters" ln="70" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'exitGame' should be composed only of lowercase letters" ln="70" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable &quot;exitGame&quot; was not initialized when declared" ln="70" sev="2" auth="devtest" rule="JSF-142\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable &quot;exitGame&quot; was not initialized when declared" ln="70" sev="3" auth="devtest" rule="INIT-03" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use of variable &quot;exitGame&quot; with static storage duration is not allowed" ln="70" sev="3" auth="devtest" rule="HICPP-3\_3\_1-a" tool="c++test" cat="HICPP-3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="4" locEndLn="70" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="70" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="70" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="70" locStartPos="0" locEndLn="70" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider using 'constexpr' for the 'GameLogic\_Thread' function" ln="72" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_2-b" tool="c++test" cat="AUTOSAR-A7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider using 'constexpr' for the 'GameLogic\_Thread' function" ln="72" sev="4" auth="devtest" rule="CODSTA-MCPP-11\_b\_cpp11" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has Cyclomatic Complexity value: 1" ln="72" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has Essential Complexity value: 1" ln="72" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has external linkage and is not declared in the header" ln="72" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has external linkage and is not declared in the header" ln="72" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has external linkage and is not declared in the header" ln="72" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has external linkage and is not declared in the header" ln="72" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has external linkage and is not declared in the header" ln="72" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has external linkage and is not declared in the header" ln="72" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' has external linkage and is not declared in the header" ln="72" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' returns a pointer type" ln="72" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'GameLogic\_Thread' returns a pointer type" ln="72" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'GameLogic\_Thread' is declared in global namespace" ln="72" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'GameLogic\_Thread' is declared in global namespace" ln="72" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'GameLogic\_Thread' is declared in global namespace" ln="72" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'GameLogic\_Thread' is declared in global namespace" ln="72" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'GameLogic\_Thread' function should be declared 'noexcept'" ln="72" sev="2" auth="devtest" rule="AUTOSAR-A15\_4\_4-a" tool="c++test" cat="AUTOSAR-A15\_4\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'GameLogic\_Thread' function should be declared 'noexcept'" ln="72" sev="3" auth="devtest" rule="CODSTA-MCPP-09" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'GameLogic\_Thread' function should be preceded by a comment that contains the '@brief' tag" ln="72" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'GameLogic\_Thread' function should be preceded by a comment that contains the '@brief' tag" ln="72" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'GameLogic\_Thread' function should be preceded by a comment that contains the '@return' tag" ln="72" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'GameLogic\_Thread' function should be preceded by a comment that contains the '@return' tag" ln="72" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'GameLogic\_Thread' function is not preceded by a comment" ln="72" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'GameLogic\_Thread' function is not preceded by a comment" ln="72" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The incorrect global function name GameLogic\_Thread was found" ln="72" sev="3" auth="devtest" rule="NAMING-34" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'GameLogic\_Thread' should be composed only of lowercase letters" ln="72" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'GameLogic\_Thread' should be composed only of lowercase letters" ln="72" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared" ln="72" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="23" locEndLn="72" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer type is declared" ln="72" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="23" locEndLn="72" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The return type of the 'GameLogic\_Thread' function should be declared as 'auto'" ln="72" sev="2" auth="devtest" rule="CODSTA-MCPP-08\_b" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="72" locStartPos="6" locEndLn="72" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<DupViol msg="Duplicated function: 'THREAD\_RETURN\_TYPE GameLogic\_Thread ( void \* ) { return (...'" ln="73" NvType="1" sev="2" auth="devtest" rule="CDD-DUPM" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="-1769734618" NvActs="3" locStartln="73" locStartPos="0" locEndLn="76" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<ElDescList>

<ElDesc srcRngStartln="73" srcRngStartPos="0" srcRngEndLn="76" srcRngEndPos="1" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="73" ElType="" desc="[Line 73] Duplicated function in file 'DeadLock.cpp'" sourceRngHash="286473175">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="79" srcRngStartPos="0" srcRngEndLn="82" srcRngEndPos="1" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="79" ElType="" desc="[Line 79] Duplicated function in file 'DeadLock.cpp'" sourceRngHash="-13596395">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Non-ascii tab found" ln="74" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="74" locStartPos="0" locEndLn="74" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="74" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="74" locStartPos="0" locEndLn="74" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="74" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="74" locStartPos="0" locEndLn="74" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="75" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="1" locEndLn="75" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast is used" ln="75" sev="3" auth="devtest" rule="HICPP-5\_4\_1-a" tool="c++test" cat="HICPP-5\_4\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast is used" ln="75" sev="3" auth="devtest" rule="CODSTA-CPP-11" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast to 'void \*' type is used" ln="75" sev="2" auth="devtest" rule="JSF-185" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast to 'void \*' type is used" ln="75" sev="2" auth="devtest" rule="AUTOSAR-A5\_2\_2-a" tool="c++test" cat="AUTOSAR-A5\_2\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast to 'void \*' type is used" ln="75" sev="2" auth="devtest" rule="MISRA2008-5\_2\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast to 'void \*' type is used" ln="75" sev="3" auth="devtest" rule="CODSTA-CPP-66" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Cast to primitive type should not be used if possible" ln="75" sev="3" auth="devtest" rule="HICPP-4\_2\_2-b" tool="c++test" cat="HICPP-4\_2\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Cast to primitive type should not be used if possible" ln="75" sev="4" auth="devtest" rule="JSF-183" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Cast to primitive type should not be used if possible" ln="75" sev="3" auth="devtest" rule="CODSTA-31" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not convert an object with integer type 'int' to an object with pointer type 'void \*'" ln="75" sev="2" auth="devtest" rule="MISRA2008-5\_2\_8" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not convert an object with integer type 'int' to an object with pointer type 'void \*'" ln="75" sev="3" auth="devtest" rule="CODSTA-65" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not convert an object with integer type 'int' to an object with pointer type 'void \*'" ln="75" sev="2" auth="devtest" rule="AUTOSAR-M5\_2\_8-a" tool="c++test" cat="AUTOSAR-M5\_2\_8" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="8" locEndLn="75" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<DupViol msg="Duplicated code: 'return (THREAD\_RETURN\_TYPE)0;'" ln="75" NvType="1" sev="3" auth="devtest" rule="CDD-DUPC" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="-1769734618" NvActs="3" locStartln="75" locStartPos="1" locEndLn="75" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<ElDescList>

<ElDesc srcRngStartln="75" srcRngStartPos="1" srcRngEndLn="75" srcRngEndPos="30" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="75" ElType="" desc="[Line 75] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="364761864">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="81" srcRngStartPos="1" srcRngEndLn="81" srcRngEndPos="30" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="81" ElType="" desc="[Line 81] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="364761864">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Non-ascii tab found" ln="75" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="0" locEndLn="75" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="75" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="0" locEndLn="75" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="75" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="0" locEndLn="75" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="75" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="15" locEndLn="75" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="75" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="15" locEndLn="75" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="75" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="75" locStartPos="15" locEndLn="75" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider using 'constexpr' for the 'Controller\_Thread' function" ln="78" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_2-b" tool="c++test" cat="AUTOSAR-A7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider using 'constexpr' for the 'Controller\_Thread' function" ln="78" sev="4" auth="devtest" rule="CODSTA-MCPP-11\_b\_cpp11" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has Cyclomatic Complexity value: 1" ln="78" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has Essential Complexity value: 1" ln="78" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has external linkage and is not declared in the header" ln="78" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has external linkage and is not declared in the header" ln="78" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has external linkage and is not declared in the header" ln="78" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has external linkage and is not declared in the header" ln="78" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has external linkage and is not declared in the header" ln="78" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has external linkage and is not declared in the header" ln="78" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' has external linkage and is not declared in the header" ln="78" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' returns a pointer type" ln="78" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Controller\_Thread' returns a pointer type" ln="78" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'Controller\_Thread' is declared in global namespace" ln="78" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'Controller\_Thread' is declared in global namespace" ln="78" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'Controller\_Thread' is declared in global namespace" ln="78" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'Controller\_Thread' is declared in global namespace" ln="78" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Controller\_Thread' function should be declared 'noexcept'" ln="78" sev="2" auth="devtest" rule="AUTOSAR-A15\_4\_4-a" tool="c++test" cat="AUTOSAR-A15\_4\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Controller\_Thread' function should be declared 'noexcept'" ln="78" sev="3" auth="devtest" rule="CODSTA-MCPP-09" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Controller\_Thread' function should be preceded by a comment that contains the '@brief' tag" ln="78" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Controller\_Thread' function should be preceded by a comment that contains the '@brief' tag" ln="78" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Controller\_Thread' function should be preceded by a comment that contains the '@return' tag" ln="78" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Controller\_Thread' function should be preceded by a comment that contains the '@return' tag" ln="78" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'Controller\_Thread' function is not preceded by a comment" ln="78" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'Controller\_Thread' function is not preceded by a comment" ln="78" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The incorrect global function name Controller\_Thread was found" ln="78" sev="3" auth="devtest" rule="NAMING-34" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'Controller\_Thread' should be composed only of lowercase letters" ln="78" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'Controller\_Thread' should be composed only of lowercase letters" ln="78" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared" ln="78" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="24" locEndLn="78" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer type is declared" ln="78" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="24" locEndLn="78" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The return type of the 'Controller\_Thread' function should be declared as 'auto'" ln="78" sev="2" auth="devtest" rule="CODSTA-MCPP-08\_b" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="78" locStartPos="6" locEndLn="78" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="80" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="80" locStartPos="0" locEndLn="80" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="80" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="80" locStartPos="0" locEndLn="80" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="80" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="80" locStartPos="0" locEndLn="80" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="81" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="1" locEndLn="81" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast is used" ln="81" sev="3" auth="devtest" rule="HICPP-5\_4\_1-a" tool="c++test" cat="HICPP-5\_4\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast is used" ln="81" sev="3" auth="devtest" rule="CODSTA-CPP-11" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast to 'void \*' type is used" ln="81" sev="2" auth="devtest" rule="JSF-185" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast to 'void \*' type is used" ln="81" sev="2" auth="devtest" rule="AUTOSAR-A5\_2\_2-a" tool="c++test" cat="AUTOSAR-A5\_2\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast to 'void \*' type is used" ln="81" sev="2" auth="devtest" rule="MISRA2008-5\_2\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="C-style cast to 'void \*' type is used" ln="81" sev="3" auth="devtest" rule="CODSTA-CPP-66" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Cast to primitive type should not be used if possible" ln="81" sev="3" auth="devtest" rule="HICPP-4\_2\_2-b" tool="c++test" cat="HICPP-4\_2\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Cast to primitive type should not be used if possible" ln="81" sev="4" auth="devtest" rule="JSF-183" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Cast to primitive type should not be used if possible" ln="81" sev="3" auth="devtest" rule="CODSTA-31" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not convert an object with integer type 'int' to an object with pointer type 'void \*'" ln="81" sev="2" auth="devtest" rule="MISRA2008-5\_2\_8" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not convert an object with integer type 'int' to an object with pointer type 'void \*'" ln="81" sev="3" auth="devtest" rule="CODSTA-65" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not convert an object with integer type 'int' to an object with pointer type 'void \*'" ln="81" sev="2" auth="devtest" rule="AUTOSAR-M5\_2\_8-a" tool="c++test" cat="AUTOSAR-M5\_2\_8" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="8" locEndLn="81" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="81" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="0" locEndLn="81" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="81" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="0" locEndLn="81" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="81" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="0" locEndLn="81" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="81" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="15" locEndLn="81" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="81" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="15" locEndLn="81" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="81" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="81" locStartPos="15" locEndLn="81" locEndPos="16" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'velocityArray'" ln="86" sev="3" auth="devtest" rule="JSF-207" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'velocityArray'" ln="86" sev="3" auth="devtest" rule="CODSTA-CPP-82" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declaration of variable 'velocityArray' contains more than one level of pointer indirection" ln="86" sev="3" auth="devtest" rule="HICPP-8\_1\_1-a" tool="c++test" cat="HICPP-8\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declaration of variable 'velocityArray' contains more than one level of pointer indirection" ln="86" sev="3" auth="devtest" rule="CODSTA-89" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'velocityArray' has external linkage and is not declared in the header" ln="86" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'velocityArray' has external linkage and is not declared in the header" ln="86" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'velocityArray' has external linkage and is not declared in the header" ln="86" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'velocityArray' has external linkage and is not declared in the header" ln="86" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'velocityArray' has external linkage and is not declared in the header" ln="86" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'velocityArray' has external linkage and is not declared in the header" ln="86" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'velocityArray' has external linkage and is not declared in the header" ln="86" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'velocityArray' is declared" ln="86" sev="5" auth="devtest" rule="MISRA-022" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: velocityArray" ln="86" sev="3" auth="devtest" rule="NAMING-18" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="86" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="0" locEndLn="86" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="86" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="0" locEndLn="86" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="86" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="0" locEndLn="86" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'velocityArray' array should not be used" ln="86" sev="3" auth="devtest" rule="STL-37" tool="c++test" cat="STL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'velocityArray' array should not be used" ln="86" sev="2" auth="devtest" rule="AUTOSAR-A18\_1\_1-a" tool="c++test" cat="AUTOSAR-A18\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'velocityArray' identifier should have the 'rg' prefix" ln="86" sev="3" auth="devtest" rule="NAMING-HN-01" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'velocityArray' variable has only one use" ln="86" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'velocityArray' variable has only one use" ln="86" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'velocityArray' variable has only one use" ln="86" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'velocityArray' variable should be commented" ln="86" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'velocityArray' variable should be commented" ln="86" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'velocityArray' should be composed only of lowercase letters" ln="86" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'velocityArray' should be composed only of lowercase letters" ln="86" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operator '\*', used for 'velocityArray' declaration, should be directly connected with the type" ln="86" sev="3" auth="devtest" rule="JSF-062" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="7" locEndLn="86" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operator '\*', used for 'velocityArray' declaration, should be directly connected with the type" ln="86" sev="4" auth="devtest" rule="FORMAT-32" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="7" locEndLn="86" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: velocityArray" ln="86" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: velocityArray" ln="86" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use of variable &quot;velocityArray&quot; with static storage duration is not allowed" ln="86" sev="3" auth="devtest" rule="HICPP-3\_3\_1-a" tool="c++test" cat="HICPP-3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="86" locStartPos="8" locEndLn="86" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has Cyclomatic Complexity value: 3" ln="88" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has Essential Complexity value: 1" ln="88" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has external linkage and is not declared in the header" ln="88" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has external linkage and is not declared in the header" ln="88" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has external linkage and is not declared in the header" ln="88" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has external linkage and is not declared in the header" ln="88" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has external linkage and is not declared in the header" ln="88" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has external linkage and is not declared in the header" ln="88" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' has external linkage and is not declared in the header" ln="88" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' returns a pointer type" ln="88" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Physics\_Thread' returns a pointer type" ln="88" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="88" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="0" locEndLn="88" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="88" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="0" locEndLn="88" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="88" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="0" locEndLn="88" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Physics\_Thread' function should be preceded by a comment that contains the '@brief' tag" ln="88" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Physics\_Thread' function should be preceded by a comment that contains the '@brief' tag" ln="88" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Physics\_Thread' function should be preceded by a comment that contains the '@return' tag" ln="88" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Physics\_Thread' function should be preceded by a comment that contains the '@return' tag" ln="88" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'Physics\_Thread' function is not preceded by a comment" ln="88" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'Physics\_Thread' function is not preceded by a comment" ln="88" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The incorrect global function name Physics\_Thread was found" ln="88" sev="3" auth="devtest" rule="NAMING-34" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'Physics\_Thread' should be composed only of lowercase letters" ln="88" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'Physics\_Thread' should be composed only of lowercase letters" ln="88" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="7" locEndLn="88" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared" ln="88" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="22" locEndLn="88" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer type is declared" ln="88" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="88" locStartPos="22" locEndLn="88" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="89" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="89" locStartPos="0" locEndLn="89" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="89" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="89" locStartPos="0" locEndLn="89" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="89" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="89" locStartPos="0" locEndLn="89" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 13" ln="89" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="89" locStartPos="0" locEndLn="89" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="A loop does not have a fixed upper nor lower bound" ln="90" sev="3" auth="devtest" rule="CODSTA-83" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="2" locEndLn="90" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="In 'Physics\_Thread' function use positive logic rather than negative logic whenever possible" ln="90" sev="5" auth="devtest" rule="CODSTA-46" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="9" locEndLn="90" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="90" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="90" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="90" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="90" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="1" locEndLn="90" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="90" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="1" locEndLn="90" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="90" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="1" locEndLn="90" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="90" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="90" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="90" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="90" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="90" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="90" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="0" locEndLn="90" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="90" sev="3" auth="devtest" rule="CODSTA-CPP-67" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="9" locEndLn="90" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="90" sev="2" auth="devtest" rule="AUTOSAR-M5\_3\_1-a" tool="c++test" cat="AUTOSAR-M5\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="9" locEndLn="90" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="90" sev="2" auth="devtest" rule="MISRA2008-5\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="9" locEndLn="90" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' is not 'effectively Boolean'" ln="90" sev="3" auth="devtest" rule="MISRA2004-12\_6\_a" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="90" locStartPos="9" locEndLn="90" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'i' shall be declared as unsigned int or signed int" ln="91" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="11" locEndLn="91" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="A loop does not have a fixed upper nor lower bound" ln="91" sev="3" auth="devtest" rule="CODSTA-83" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="3" locEndLn="91" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Between conditional statement 'for' and its opening parenthesis '(' should be exactly one space" ln="91" sev="3" auth="devtest" rule="FORMAT-12" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="3" locEndLn="91" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<DupViol msg="Duplicated code: 'for(int i = 0;'" ln="91" NvType="1" sev="3" auth="devtest" rule="CDD-DUPC" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="-1769734618" NvActs="3" locStartln="91" locStartPos="3" locEndLn="91" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<ElDescList>

<ElDesc srcRngStartln="91" srcRngStartPos="3" srcRngEndLn="91" srcRngEndPos="17" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="" desc="[Line 91] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="1701351733">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="122" srcRngStartPos="3" srcRngEndLn="122" srcRngEndPos="17" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="122" ElType="" desc="[Line 122] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="1701351733">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="1" srcRngEndLn="9" srcRngEndPos="16" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="" desc="[Line 9] Duplicated code in file 'MemoryLeak.cpp'" sourceRngHash="2017663175">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Non-ascii tab found" ln="91" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="91" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="91" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="91" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="1" locEndLn="91" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="91" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="1" locEndLn="91" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="91" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="1" locEndLn="91" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="91" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="2" locEndLn="91" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="91" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="2" locEndLn="91" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="91" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="2" locEndLn="91" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="91" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="91" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Postfix operator applied to variable 'i'; prefer prefix type" ln="91" sev="3" auth="devtest" rule="OPT-04" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="41" locEndLn="91" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="91" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="91" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="91" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="91" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="0" locEndLn="91" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'i' variable should be commented" ln="91" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="11" locEndLn="91" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'i' variable should be commented" ln="91" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="11" locEndLn="91" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="91" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="91" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="91" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="91" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="91" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="91" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="91" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="91" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'i' variable should contain a braced initializer" ln="91" sev="2" auth="devtest" rule="AUTOSAR-A8\_5\_2-a" tool="c++test" cat="AUTOSAR-A8\_5\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="11" locEndLn="91" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'i' variable should contain a braced initializer" ln="91" sev="3" auth="devtest" rule="CODSTA-MCPP-38" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="11" locEndLn="91" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="91" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="91" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="7" locEndLn="91" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Variable &quot;i&quot; declaration was found in control statement" ln="91" sev="3" auth="devtest" rule="OPT-10" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="91" locStartPos="11" locEndLn="91" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<DupViol msg="Duplicated code: 'LOCK\_ACQUIRE(changePositionMutex);'" ln="92" NvType="1" sev="3" auth="devtest" rule="CDD-DUPC" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="-1769734618" NvActs="3" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="38" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<ElDescList>

<ElDesc srcRngStartln="92" srcRngStartPos="4" srcRngEndLn="92" srcRngEndPos="38" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType="" desc="[Line 92] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="543344818">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="118" srcRngStartPos="3" srcRngEndLn="118" srcRngEndPos="37" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="118" ElType="" desc="[Line 118] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="543344818">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Non-ascii tab found" ln="92" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="0" locEndLn="92" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="0" locEndLn="92" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="0" locEndLn="92" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="1" locEndLn="92" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="1" locEndLn="92" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="1" locEndLn="92" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="2" locEndLn="92" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="2" locEndLn="92" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="2" locEndLn="92" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="3" locEndLn="92" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="3" locEndLn="92" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="92" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="3" locEndLn="92" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_mutex\_lock' is called without scope resolution operator '::'" ln="92" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="92" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="92" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="92" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="92" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="92" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="92" locStartPos="4" locEndLn="92" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<FlowViol msg="&quot;participants&quot; is used in two critical sections in context of single method, using one critical section will improve atomicity of operation" ln="93" ruleSAFMsg="Usage of &quot;participants&quot; in second critical section" auth="devtest" sev="2" rule="BD-TRS-DIFCS" ruleSCSCMsg="Usage of &quot;participants&quot; in first critical section" pkg="Physics" tool="c++test" id="208209936" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="0" locEndLn="94" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="93" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="94" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props>

<Prop key="Tracked variables" val="Variable used in critical section"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="90" srcRngStartPos="0" srcRngEndLn="91" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="90" ElType="." desc="while (!exitGame) {" rngLn="90">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType=".C" desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Usage of &quot;participants&quot; in first critical section" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType="." desc="position.translate(\*velocityArray[i]);" rngLn="94">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="ring.contains(...)" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="getPosition()" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="point.squareDistanceTo(...)" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="21" srcRngStartPos="0" srcRngEndLn="22" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="21" ElType="." desc="return ((\_x - point.\_x) \* (\_x - point.\_x)) + ((\_y - point.\_y) \* (\_y - point.\_y));" rngLn="21">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="return point.squareDistanceTo(getPosition()) &lt;= (\_radius \* \_radius);" rngLn="29">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="assertion(ring.contains(position), &quot;Participant is out of ring&quot;);" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="60" srcRngStartPos="0" srcRngEndLn="61" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="60" ElType="." desc="if (!condition) {" rngLn="60">

<Props/>

<Anns>

<Ann msg="Condition evaluation: !condition (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="98" srcRngStartPos="0" srcRngEndLn="99" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="98" ElType="!" desc="LOCK\_RELEASE(changePositionMutex);" rngLn="98">

<Props/>

<Anns>

<Ann msg="Unlocking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType=".P" desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<Props/>

<Anns>

<Ann msg="Usage of &quot;participants&quot; in second critical section" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Non-ascii tab found" ln="93" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="0" locEndLn="93" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="0" locEndLn="93" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="0" locEndLn="93" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="1" locEndLn="93" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="1" locEndLn="93" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="1" locEndLn="93" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="2" locEndLn="93" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="2" locEndLn="93" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="2" locEndLn="93" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="3" locEndLn="93" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="3" locEndLn="93" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="93" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="3" locEndLn="93" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'position' variable should be commented" ln="93" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="11" locEndLn="93" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'position' variable should be commented" ln="93" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="11" locEndLn="93" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'position' variable should be declared with the 'auto&amp;' type specifier" ln="93" sev="2" auth="devtest" rule="CODSTA-MCPP-08\_a" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="11" locEndLn="93" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'position' variable should contain a braced initializer" ln="93" sev="2" auth="devtest" rule="AUTOSAR-A8\_5\_2-a" tool="c++test" cat="AUTOSAR-A8\_5\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="11" locEndLn="93" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'position' variable should contain a braced initializer" ln="93" sev="3" auth="devtest" rule="CODSTA-MCPP-38" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="11" locEndLn="93" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Variable &quot;position&quot; declaration was found in control statement" ln="93" sev="3" auth="devtest" rule="OPT-10" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="11" locEndLn="93" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Variable 'position' should be declared using 'auto' specifier" ln="93" sev="3" auth="devtest" rule="HICPP-7\_1\_8-a" tool="c++test" cat="HICPP-7\_1\_8" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="93" locStartPos="11" locEndLn="93" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<FlowViol msg="&quot;velocityArray&quot; is used in two critical sections in context of single method, using one critical section will improve atomicity of operation" ln="94" ruleSAFMsg="Usage of &quot;velocityArray&quot; in second critical section" auth="devtest" sev="2" rule="BD-TRS-DIFCS" ruleSCSCMsg="Usage of &quot;velocityArray&quot; in first critical section" pkg="Physics" tool="c++test" id="1066261748" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="0" locEndLn="95" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="94" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="95" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props>

<Prop key="Tracked variables" val="Variable used in critical section"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="90" srcRngStartPos="0" srcRngEndLn="91" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="90" ElType="." desc="while (!exitGame) {" rngLn="90">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType="." desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType=".C" desc="position.translate(\*velocityArray[i]);" rngLn="94">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Usage of &quot;velocityArray&quot; in first critical section" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="ring.contains(...)" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="getPosition()" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="point.squareDistanceTo(...)" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="21" srcRngStartPos="0" srcRngEndLn="22" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="21" ElType="." desc="return ((\_x - point.\_x) \* (\_x - point.\_x)) + ((\_y - point.\_y) \* (\_y - point.\_y));" rngLn="21">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="return point.squareDistanceTo(getPosition()) &lt;= (\_radius \* \_radius);" rngLn="29">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="assertion(ring.contains(position), &quot;Participant is out of ring&quot;);" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="60" srcRngStartPos="0" srcRngEndLn="61" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="60" ElType="." desc="if (!condition) {" rngLn="60">

<Props/>

<Anns>

<Ann msg="Condition evaluation: !condition (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="98" srcRngStartPos="0" srcRngEndLn="99" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="98" ElType="!" desc="LOCK\_RELEASE(changePositionMutex);" rngLn="98">

<Props/>

<Anns>

<Ann msg="Unlocking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType="." desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType=".P" desc="position.translate(\*velocityArray[i]);" rngLn="94">

<Props/>

<Anns>

<Ann msg="Usage of &quot;velocityArray&quot; in second critical section" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Dereferenced pointer type expression should not be used" ln="94" sev="3" auth="devtest" rule="PB-01" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="24" locEndLn="94" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="0" locEndLn="94" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="0" locEndLn="94" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="0" locEndLn="94" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="1" locEndLn="94" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="1" locEndLn="94" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="1" locEndLn="94" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="2" locEndLn="94" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="2" locEndLn="94" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="2" locEndLn="94" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="3" locEndLn="94" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="3" locEndLn="94" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="94" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="94" locStartPos="3" locEndLn="94" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<FlowViol msg="&quot;ring&quot; is used in two critical sections in context of single method, using one critical section will improve atomicity of operation" ln="95" ruleSAFMsg="Usage of &quot;ring&quot; in second critical section" auth="devtest" sev="2" rule="BD-TRS-DIFCS" ruleSCSCMsg="Usage of &quot;ring&quot; in first critical section" pkg="Physics" tool="c++test" id="-1966185267" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="0" locEndLn="96" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="95" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="96" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props>

<Prop key="Tracked variables" val="Variable used in critical section"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="90" srcRngStartPos="0" srcRngEndLn="91" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="90" ElType="." desc="while (!exitGame) {" rngLn="90">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType="." desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType="." desc="position.translate(\*velocityArray[i]);" rngLn="94">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType=".C" desc="ring.contains(...)" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="getPosition()" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="point.squareDistanceTo(...)" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="21" srcRngStartPos="0" srcRngEndLn="22" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="21" ElType="." desc="return ((\_x - point.\_x) \* (\_x - point.\_x)) + ((\_y - point.\_y) \* (\_y - point.\_y));" rngLn="21">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="return point.squareDistanceTo(getPosition()) &lt;= (\_radius \* \_radius);" rngLn="29">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Usage of &quot;ring&quot; in first critical section" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="assertion(ring.contains(position), &quot;Participant is out of ring&quot;);" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="60" srcRngStartPos="0" srcRngEndLn="61" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="60" ElType="." desc="if (!condition) {" rngLn="60">

<Props/>

<Anns>

<Ann msg="Condition evaluation: !condition (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="98" srcRngStartPos="0" srcRngEndLn="99" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="98" ElType="!" desc="LOCK\_RELEASE(changePositionMutex);" rngLn="98">

<Props/>

<Anns>

<Ann msg="Unlocking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType="." desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType="." desc="position.translate(\*velocityArray[i]);" rngLn="94">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType=".P" desc="ring.contains(...)" rngLn="95">

<Props/>

<Anns>

<Ann msg="Usage of &quot;ring&quot; in second critical section" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Non-ascii tab found" ln="95" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="0" locEndLn="95" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="0" locEndLn="95" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="0" locEndLn="95" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="1" locEndLn="95" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="1" locEndLn="95" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="1" locEndLn="95" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="2" locEndLn="95" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="2" locEndLn="95" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="2" locEndLn="95" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="3" locEndLn="95" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="3" locEndLn="95" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="95" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="3" locEndLn="95" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'assertion' is called without scope resolution operator '::'" ln="95" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="4" locEndLn="95" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: Participant is out of ring" ln="95" sev="5" auth="devtest" rule="CWE-798-a" tool="c++test" cat="CWE-798" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="39" locEndLn="95" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: Participant is out of ring" ln="95" sev="5" auth="devtest" rule="CODSTA-203" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="39" locEndLn="95" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: Participant is out of ring" ln="95" sev="1" auth="devtest" rule="APSC\_DV-003110-a" tool="c++test" cat="APSC\_DV-003110" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="95" locStartPos="39" locEndLn="95" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: Participant is out of ring" ln="95" sev="1" auth="devtest" rule="CERT\_C-MSC41-a" tool="c++test" cat="CERT\_C-MSC41" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="39" locEndLn="95" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The string literal should not be passed as an argument of the 'const char \*' type in the 'assertion' function call" ln="95" sev="2" auth="devtest" rule="AUTOSAR-A27\_0\_4-d" tool="c++test" cat="AUTOSAR-A27\_0\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="39" locEndLn="95" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The string literal should not be passed as an argument of the 'const char \*' type in the 'assertion' function call" ln="95" sev="3" auth="devtest" rule="PB-76" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="95" locStartPos="39" locEndLn="95" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="0" locEndLn="96" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="0" locEndLn="96" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="0" locEndLn="96" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="1" locEndLn="96" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="1" locEndLn="96" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="1" locEndLn="96" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="2" locEndLn="96" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="2" locEndLn="96" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="2" locEndLn="96" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="3" locEndLn="96" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="3" locEndLn="96" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="3" locEndLn="96" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="4" locEndLn="96" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="4" locEndLn="96" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="4" locEndLn="96" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="5" locEndLn="96" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="5" locEndLn="96" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="5" locEndLn="96" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="6" locEndLn="96" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="6" locEndLn="96" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="6" locEndLn="96" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="7" locEndLn="96" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="7" locEndLn="96" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="7" locEndLn="96" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="8" locEndLn="96" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="8" locEndLn="96" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="8" locEndLn="96" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="9" locEndLn="96" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="9" locEndLn="96" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="9" locEndLn="96" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="10" locEndLn="96" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="10" locEndLn="96" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="10" locEndLn="96" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="11" locEndLn="96" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="11" locEndLn="96" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="96" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="96" locStartPos="11" locEndLn="96" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="0" locEndLn="97" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="0" locEndLn="97" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="0" locEndLn="97" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="1" locEndLn="97" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="1" locEndLn="97" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="1" locEndLn="97" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="2" locEndLn="97" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="2" locEndLn="97" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="2" locEndLn="97" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="3" locEndLn="97" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="3" locEndLn="97" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="3" locEndLn="97" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="4" locEndLn="97" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="4" locEndLn="97" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="4" locEndLn="97" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="5" locEndLn="97" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="5" locEndLn="97" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="5" locEndLn="97" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="6" locEndLn="97" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="6" locEndLn="97" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="6" locEndLn="97" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="7" locEndLn="97" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="7" locEndLn="97" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="7" locEndLn="97" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="8" locEndLn="97" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="8" locEndLn="97" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="8" locEndLn="97" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="9" locEndLn="97" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="9" locEndLn="97" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="9" locEndLn="97" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="10" locEndLn="97" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="10" locEndLn="97" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="10" locEndLn="97" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="11" locEndLn="97" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="11" locEndLn="97" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="97" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="97" locStartPos="11" locEndLn="97" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="0" locEndLn="98" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="0" locEndLn="98" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="0" locEndLn="98" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="1" locEndLn="98" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="1" locEndLn="98" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="1" locEndLn="98" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="2" locEndLn="98" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="2" locEndLn="98" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="2" locEndLn="98" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="3" locEndLn="98" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="3" locEndLn="98" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="98" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="3" locEndLn="98" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_mutex\_unlock' is called without scope resolution operator '::'" ln="98" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="98" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="98" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="98" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="98" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="98" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="98" locStartPos="4" locEndLn="98" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="0" locEndLn="99" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="0" locEndLn="99" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="0" locEndLn="99" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="1" locEndLn="99" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="1" locEndLn="99" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="1" locEndLn="99" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="2" locEndLn="99" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="2" locEndLn="99" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="99" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="99" locStartPos="2" locEndLn="99" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="A 'U' suffix shall be applied to constant: 20" ln="100" sev="3" auth="devtest" rule="HICPP-4\_2\_1-a" tool="c++test" cat="HICPP-4\_2\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="9" locEndLn="100" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<DupViol msg="Duplicated code: 'SLEEP(STEP);'" ln="100" NvType="1" sev="3" auth="devtest" rule="CDD-DUPC" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="-1769734618" NvActs="3" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<ElDescList>

<ElDesc srcRngStartln="100" srcRngStartPos="3" srcRngEndLn="100" srcRngEndPos="15" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="100" ElType="" desc="[Line 100] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="-847978353">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="128" srcRngStartPos="3" srcRngEndLn="128" srcRngEndPos="15" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="128" ElType="" desc="[Line 128] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="-847978353">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Implicit conversion between signed and unsigned type in the function call on argument '1' shall not be used" ln="100" sev="2" auth="devtest" rule="MISRA2008-5\_0\_4\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="9" locEndLn="100" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion between signed and unsigned type in the function call on argument '1' shall not be used" ln="100" sev="2" auth="devtest" rule="AUTOSAR-M5\_0\_4-a" tool="c++test" cat="AUTOSAR-M5\_0\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="9" locEndLn="100" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion between signed and unsigned type in the function call on argument '1' shall not be used" ln="100" sev="3" auth="devtest" rule="MISRA2004-10\_1\_a" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="9" locEndLn="100" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="0" locEndLn="100" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="0" locEndLn="100" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="0" locEndLn="100" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="1" locEndLn="100" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="1" locEndLn="100" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="1" locEndLn="100" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="2" locEndLn="100" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="2" locEndLn="100" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="100" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="2" locEndLn="100" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'sleep' is called without scope resolution operator '::'" ln="100" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The type 'int' of function argument number '1' does not match declared type 'unsigned int'" ln="100" sev="3" auth="devtest" rule="PB-11" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="100" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'sleep' return value" ln="100" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'sleep' return value" ln="100" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'sleep' return value" ln="100" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'sleep' return value" ln="100" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="100" locStartPos="3" locEndLn="100" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="101" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="101" locStartPos="0" locEndLn="101" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="101" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="101" locStartPos="0" locEndLn="101" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="101" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="101" locStartPos="0" locEndLn="101" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="101" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="101" locStartPos="1" locEndLn="101" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="101" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="101" locStartPos="1" locEndLn="101" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="101" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="101" locStartPos="1" locEndLn="101" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="102" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="2" locEndLn="102" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<DupViol msg="Duplicated code: 'return 0;'" ln="102" NvType="1" sev="3" auth="devtest" rule="CDD-DUPC" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="-1769734618" NvActs="3" locStartln="102" locStartPos="2" locEndLn="102" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<ElDescList>

<ElDesc srcRngStartln="102" srcRngStartPos="2" srcRngEndLn="102" srcRngEndPos="11" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="102" ElType="" desc="[Line 102] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="-926747909">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="130" srcRngStartPos="2" srcRngEndLn="130" srcRngEndPos="11" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="130" ElType="" desc="[Line 130] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="-926747909">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="13" srcRngStartPos="1" srcRngEndLn="13" srcRngEndPos="10" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="13" ElType="" desc="[Line 13] Duplicated code in file 'NullPointer.cpp'" sourceRngHash="-926747909">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="19" srcRngStartPos="29" srcRngEndLn="19" srcRngEndPos="38" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="" desc="[Line 19] Duplicated code in file 'Shapes.hpp'" sourceRngHash="-926747909">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Function with pointer return type returns '0'" ln="102" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="102" sev="2" auth="devtest" rule="AUTOSAR-M4\_10\_2-a" tool="c++test" cat="AUTOSAR-M4\_10\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="102" sev="2" auth="devtest" rule="MISRA2008-4\_10\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="102" sev="3" auth="devtest" rule="CODSTA-CPP-63" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="102" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-a" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="102" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="102" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="102" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="102" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="0" locEndLn="102" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="102" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="0" locEndLn="102" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="102" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="0" locEndLn="102" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="102" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="1" locEndLn="102" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="102" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="1" locEndLn="102" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="102" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="1" locEndLn="102" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="102" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="102" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="102" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="102" locStartPos="9" locEndLn="102" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="103" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="103" locStartPos="0" locEndLn="103" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="103" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="103" locStartPos="0" locEndLn="103" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="103" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="103" locStartPos="0" locEndLn="103" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<FlowViol msg="Unreleased lock: changePositionMutex" ln="103" ruleSAFMsg="Point prior to which the lock is to be released" auth="devtest" sev="1" rule="BD-TRS-LOCK" ruleSCSCMsg="Point where the lock is acquired" pkg="Physics" tool="c++test" id="-1760906373" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="103" locStartPos="0" locEndLn="104" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="92" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="93" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="90" srcRngStartPos="0" srcRngEndLn="91" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="90" ElType="." desc="while (!exitGame) {" rngLn="90">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType=".C" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Point where the lock is acquired" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType="." desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType="." desc="position.translate(\*velocityArray[i]);" rngLn="94">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="ring.contains(...)" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="getPosition()" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="point.squareDistanceTo(...)" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="21" srcRngStartPos="0" srcRngEndLn="22" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="21" ElType="." desc="return ((\_x - point.\_x) \* (\_x - point.\_x)) + ((\_y - point.\_y) \* (\_y - point.\_y));" rngLn="21">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="return point.squareDistanceTo(getPosition()) &lt;= (\_radius \* \_radius);" rngLn="29">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="!E" desc="assertion(ring.contains(position), &quot;Participant is out of ring&quot;);" rngLn="95" thrownTypes="const char \*" throwingMethod="assertion">

<ElDescList>

<ElDesc srcRngStartln="60" srcRngStartPos="0" srcRngEndLn="61" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="60" ElType="." desc="if (!condition) {" rngLn="60">

<Props/>

<Anns>

<Ann msg="Condition evaluation: !condition (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="61" srcRngStartPos="0" srcRngEndLn="62" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="61" ElType="!E" desc="throw message;" rngLn="61" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="assertion() throws const char \*" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="103" srcRngStartPos="0" srcRngEndLn="104" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="103" ElType=".P" desc="}" rngLn="103">

<Props/>

<Anns>

<Ann msg="Point prior to which the lock is to be released" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Unreleased lock: changePositionMutex" ln="103" ruleSAFMsg="Point prior to which the lock is to be released" auth="devtest" sev="3" rule="CERT\_C-CON01-a" ruleSCSCMsg="Point where the lock is acquired" pkg="Physics" tool="c++test" id="-1441471257" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="103" locStartPos="0" locEndLn="104" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="92" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="93" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="90" srcRngStartPos="0" srcRngEndLn="91" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="90" ElType="." desc="while (!exitGame) {" rngLn="90">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType=".C" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Point where the lock is acquired" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType="." desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType="." desc="position.translate(\*velocityArray[i]);" rngLn="94">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="ring.contains(...)" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="getPosition()" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="point.squareDistanceTo(...)" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="21" srcRngStartPos="0" srcRngEndLn="22" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="21" ElType="." desc="return ((\_x - point.\_x) \* (\_x - point.\_x)) + ((\_y - point.\_y) \* (\_y - point.\_y));" rngLn="21">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="return point.squareDistanceTo(getPosition()) &lt;= (\_radius \* \_radius);" rngLn="29">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="!E" desc="assertion(ring.contains(position), &quot;Participant is out of ring&quot;);" rngLn="95" thrownTypes="const char \*" throwingMethod="assertion">

<ElDescList>

<ElDesc srcRngStartln="60" srcRngStartPos="0" srcRngEndLn="61" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="60" ElType="." desc="if (!condition) {" rngLn="60">

<Props/>

<Anns>

<Ann msg="Condition evaluation: !condition (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="61" srcRngStartPos="0" srcRngEndLn="62" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="61" ElType="!E" desc="throw message;" rngLn="61" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="assertion() throws const char \*" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="103" srcRngStartPos="0" srcRngEndLn="104" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="103" ElType=".P" desc="}" rngLn="103">

<Props/>

<Anns>

<Ann msg="Point prior to which the lock is to be released" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Unreleased lock: changePositionMutex" ln="103" ruleSAFMsg="Point prior to which the lock is to be released" auth="devtest" sev="4" rule="MISRA2012-DIR-4\_13\_d" ruleSCSCMsg="Point where the lock is acquired" pkg="Physics" tool="c++test" id="108210863" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="103" locStartPos="0" locEndLn="104" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="92" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="93" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="90" srcRngStartPos="0" srcRngEndLn="91" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="90" ElType="." desc="while (!exitGame) {" rngLn="90">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType=".C" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Point where the lock is acquired" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType="." desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType="." desc="position.translate(\*velocityArray[i]);" rngLn="94">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="ring.contains(...)" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="getPosition()" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="point.squareDistanceTo(...)" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="21" srcRngStartPos="0" srcRngEndLn="22" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="21" ElType="." desc="return ((\_x - point.\_x) \* (\_x - point.\_x)) + ((\_y - point.\_y) \* (\_y - point.\_y));" rngLn="21">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="return point.squareDistanceTo(getPosition()) &lt;= (\_radius \* \_radius);" rngLn="29">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="!E" desc="assertion(ring.contains(position), &quot;Participant is out of ring&quot;);" rngLn="95" thrownTypes="const char \*" throwingMethod="assertion">

<ElDescList>

<ElDesc srcRngStartln="60" srcRngStartPos="0" srcRngEndLn="61" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="60" ElType="." desc="if (!condition) {" rngLn="60">

<Props/>

<Anns>

<Ann msg="Condition evaluation: !condition (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="61" srcRngStartPos="0" srcRngEndLn="62" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="61" ElType="!E" desc="throw message;" rngLn="61" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="assertion() throws const char \*" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="103" srcRngStartPos="0" srcRngEndLn="104" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="103" ElType=".P" desc="}" rngLn="103">

<Props/>

<Anns>

<Ann msg="Point prior to which the lock is to be released" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Unreleased lock: changePositionMutex" ln="103" ruleSAFMsg="Point prior to which the lock is to be released" auth="devtest" sev="4" rule="MISRAC2012-DIR\_4\_13-d" ruleSCSCMsg="Point where the lock is acquired" pkg="Physics" tool="c++test" id="-770235758" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="103" locStartPos="0" locEndLn="104" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="92" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="93" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="90" srcRngStartPos="0" srcRngEndLn="91" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="90" ElType="." desc="while (!exitGame) {" rngLn="90">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="91" srcRngStartPos="0" srcRngEndLn="92" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="91" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="91">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="92" srcRngStartPos="0" srcRngEndLn="93" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="92" ElType=".C" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="92">

<Props/>

<Anns>

<Ann msg="Point where the lock is acquired" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="93" srcRngStartPos="0" srcRngEndLn="94" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="93" ElType="." desc="Point&amp; position = participants[i]->getPosition();" rngLn="93">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="94" srcRngStartPos="0" srcRngEndLn="95" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="94" ElType="." desc="position.translate(\*velocityArray[i]);" rngLn="94">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="." desc="ring.contains(...)" rngLn="95">

<ElDescList>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="getPosition()" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="point.squareDistanceTo(...)" rngLn="29">

<ElDescList>

<ElDesc srcRngStartln="21" srcRngStartPos="0" srcRngEndLn="22" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="21" ElType="." desc="return ((\_x - point.\_x) \* (\_x - point.\_x)) + ((\_y - point.\_y) \* (\_y - point.\_y));" rngLn="21">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="29" srcRngStartPos="0" srcRngEndLn="30" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="29" ElType="." desc="return point.squareDistanceTo(getPosition()) &lt;= (\_radius \* \_radius);" rngLn="29">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="95" srcRngStartPos="0" srcRngEndLn="96" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="95" ElType="!E" desc="assertion(ring.contains(position), &quot;Participant is out of ring&quot;);" rngLn="95" thrownTypes="const char \*" throwingMethod="assertion">

<ElDescList>

<ElDesc srcRngStartln="60" srcRngStartPos="0" srcRngEndLn="61" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="60" ElType="." desc="if (!condition) {" rngLn="60">

<Props/>

<Anns>

<Ann msg="Condition evaluation: !condition (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="61" srcRngStartPos="0" srcRngEndLn="62" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="61" ElType="!E" desc="throw message;" rngLn="61" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="assertion() throws const char \*" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="103" srcRngStartPos="0" srcRngEndLn="104" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="103" ElType=".P" desc="}" rngLn="103">

<Props/>

<Anns>

<Ann msg="Point prior to which the lock is to be released" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Consider encapsulating 'currentCameraVelocity'" ln="108" sev="3" auth="devtest" rule="JSF-207" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Consider encapsulating 'currentCameraVelocity'" ln="108" sev="3" auth="devtest" rule="CODSTA-CPP-82" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'currentCameraVelocity' has external linkage and is not declared in the header" ln="108" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'currentCameraVelocity' has external linkage and is not declared in the header" ln="108" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'currentCameraVelocity' has external linkage and is not declared in the header" ln="108" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'currentCameraVelocity' has external linkage and is not declared in the header" ln="108" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'currentCameraVelocity' has external linkage and is not declared in the header" ln="108" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'currentCameraVelocity' has external linkage and is not declared in the header" ln="108" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'currentCameraVelocity' has external linkage and is not declared in the header" ln="108" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'currentCameraVelocity' is declared" ln="108" sev="5" auth="devtest" rule="MISRA-022" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: currentCameraVelocity" ln="108" sev="3" auth="devtest" rule="NAMING-18" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: currentCameraVelocity" ln="108" sev="3" auth="devtest" rule="NAMING-22" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="108" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="0" locEndLn="108" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="108" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="0" locEndLn="108" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="108" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="0" locEndLn="108" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'currentCameraVelocity' uninitialized when declared" ln="108" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'currentCameraVelocity' identifier should have the 'p' prefix" ln="108" sev="3" auth="devtest" rule="NAMING-HN-34" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'currentCameraVelocity' variable should be commented" ln="108" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'currentCameraVelocity' variable should be commented" ln="108" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'currentCameraVelocity' should be composed only of lowercase letters" ln="108" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'currentCameraVelocity' should be composed only of lowercase letters" ln="108" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operator '\*', used for 'currentCameraVelocity' declaration, should be directly connected with the type" ln="108" sev="3" auth="devtest" rule="JSF-062" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="7" locEndLn="108" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operator '\*', used for 'currentCameraVelocity' declaration, should be directly connected with the type" ln="108" sev="4" auth="devtest" rule="FORMAT-32" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="7" locEndLn="108" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: currentCameraVelocity" ln="108" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: currentCameraVelocity" ln="108" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use of variable &quot;currentCameraVelocity&quot; with static storage duration is not allowed" ln="108" sev="3" auth="devtest" rule="HICPP-3\_3\_1-a" tool="c++test" cat="HICPP-3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="108" locStartPos="8" locEndLn="108" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'object' as const" ln="110" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'object' as const" ln="110" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'object' as const" ln="110" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'object' as const" ln="110" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare parameter 'object' as const" ln="110" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has Cyclomatic Complexity value: 1" ln="110" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has Essential Complexity value: 1" ln="110" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has external linkage and is not declared in the header" ln="110" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has external linkage and is not declared in the header" ln="110" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has external linkage and is not declared in the header" ln="110" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has external linkage and is not declared in the header" ln="110" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has external linkage and is not declared in the header" ln="110" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has external linkage and is not declared in the header" ln="110" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' has external linkage and is not declared in the header" ln="110" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' is empty" ln="110" sev="2" auth="devtest" rule="MISRA2008-0\_1\_8\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' is empty" ln="110" sev="4" auth="devtest" rule="OPT-32\_b" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'draw' is empty" ln="110" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_8-b" tool="c++test" cat="AUTOSAR-M0\_1\_8" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: draw" ln="110" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="110" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="0" locEndLn="110" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="110" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="0" locEndLn="110" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="110" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="0" locEndLn="110" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Parameter 'object' is not used" ln="110" sev="2" auth="devtest" rule="OPT-03" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Parameter 'object' is not used" ln="110" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_4-a" tool="c++test" cat="AUTOSAR-A0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Parameter 'object' is not used" ln="110" sev="3" auth="devtest" rule="OPT-31" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Parameter 'object' is not used" ln="110" sev="2" auth="devtest" rule="MISRA2008-0\_1\_11" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="R-828: parameter &quot;object&quot; was never referenced" ln="110" sev="5" auth="devtest" rule="PARSER-REMARK" tool="c++test" cat="PARSER" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Return type is not placed in line before function 'draw'" ln="110" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'const' qualifier should be placed on the right hand side of the type" ln="110" sev="3" auth="devtest" rule="FORMAT-47\_a" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="11" locEndLn="110" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'const' qualifier should be placed on the right hand side of the type" ln="110" sev="3" auth="devtest" rule="HICPP-7\_1\_4-a" tool="c++test" cat="HICPP-7\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="11" locEndLn="110" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'draw' function should be declared 'noexcept'" ln="110" sev="2" auth="devtest" rule="AUTOSAR-A15\_4\_4-a" tool="c++test" cat="AUTOSAR-A15\_4\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'draw' function should be declared 'noexcept'" ln="110" sev="3" auth="devtest" rule="CODSTA-MCPP-09" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'draw' function should be preceded by a comment that contains the '@brief' tag" ln="110" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'draw' function should be preceded by a comment that contains the '@brief' tag" ln="110" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'object' identifier should have the 'p' prefix" ln="110" sev="3" auth="devtest" rule="NAMING-HN-34" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'object' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="110" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'object' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="110" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'draw' function is not preceded by a comment" ln="110" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'draw' function is not preceded by a comment" ln="110" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="6" locEndLn="110" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operator '\*', used for 'object' declaration, should be directly connected with the type" ln="110" sev="3" auth="devtest" rule="JSF-062" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="23" locEndLn="110" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operator '\*', used for 'object' declaration, should be directly connected with the type" ln="110" sev="4" auth="devtest" rule="FORMAT-32" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="23" locEndLn="110" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared: object" ln="110" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer type is declared: object" ln="110" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="110" locStartPos="24" locEndLn="110" locEndPos="25" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="111" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="111" locStartPos="0" locEndLn="111" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="111" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="111" locStartPos="0" locEndLn="111" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="111" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="111" locStartPos="0" locEndLn="111" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="112" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="112" locStartPos="0" locEndLn="112" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="112" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="112" locStartPos="0" locEndLn="112" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="112" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="112" locStartPos="0" locEndLn="112" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="112" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="112" locStartPos="1" locEndLn="112" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="112" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="112" locStartPos="1" locEndLn="112" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="112" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="112" locStartPos="1" locEndLn="112" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="113" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="113" locStartPos="0" locEndLn="113" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="113" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="113" locStartPos="0" locEndLn="113" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="113" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="113" locStartPos="0" locEndLn="113" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has Cyclomatic Complexity value: 3" ln="115" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has Essential Complexity value: 1" ln="115" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has external linkage and is not declared in the header" ln="115" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has external linkage and is not declared in the header" ln="115" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has external linkage and is not declared in the header" ln="115" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has external linkage and is not declared in the header" ln="115" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has external linkage and is not declared in the header" ln="115" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has external linkage and is not declared in the header" ln="115" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' has external linkage and is not declared in the header" ln="115" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' returns a pointer type" ln="115" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'Render\_Thread' returns a pointer type" ln="115" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="115" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="0" locEndLn="115" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="115" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="0" locEndLn="115" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="115" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="0" locEndLn="115" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Render\_Thread' function should be preceded by a comment that contains the '@brief' tag" ln="115" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Render\_Thread' function should be preceded by a comment that contains the '@brief' tag" ln="115" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Render\_Thread' function should be preceded by a comment that contains the '@return' tag" ln="115" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'Render\_Thread' function should be preceded by a comment that contains the '@return' tag" ln="115" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'Render\_Thread' function is not preceded by a comment" ln="115" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'Render\_Thread' function is not preceded by a comment" ln="115" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The incorrect global function name Render\_Thread was found" ln="115" sev="3" auth="devtest" rule="NAMING-34" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'Render\_Thread' should be composed only of lowercase letters" ln="115" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'Render\_Thread' should be composed only of lowercase letters" ln="115" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="7" locEndLn="115" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared" ln="115" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="21" locEndLn="115" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The parameter of pointer type is declared" ln="115" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="115" locStartPos="21" locEndLn="115" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="116" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="116" locStartPos="0" locEndLn="116" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="116" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="116" locStartPos="0" locEndLn="116" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="116" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="116" locStartPos="0" locEndLn="116" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="116" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="116" locStartPos="0" locEndLn="116" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="A loop does not have a fixed upper nor lower bound" ln="117" sev="3" auth="devtest" rule="CODSTA-83" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="2" locEndLn="117" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="In 'Render\_Thread' function use positive logic rather than negative logic whenever possible" ln="117" sev="5" auth="devtest" rule="CODSTA-46" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="9" locEndLn="117" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="117" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="117" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="117" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="117" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="1" locEndLn="117" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="117" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="1" locEndLn="117" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="117" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="1" locEndLn="117" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="117" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="117" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="117" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="117" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="117" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="117" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="0" locEndLn="117" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="117" sev="3" auth="devtest" rule="CODSTA-CPP-67" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="9" locEndLn="117" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="117" sev="2" auth="devtest" rule="AUTOSAR-M5\_3\_1-a" tool="c++test" cat="AUTOSAR-M5\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="9" locEndLn="117" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' has 'int' type instead of 'bool' type" ln="117" sev="2" auth="devtest" rule="MISRA2008-5\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="9" locEndLn="117" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The operand of logical operator '!' is not 'effectively Boolean'" ln="117" sev="3" auth="devtest" rule="MISRA2004-12\_6\_a" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="117" locStartPos="9" locEndLn="117" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="0" locEndLn="118" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="0" locEndLn="118" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="0" locEndLn="118" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="1" locEndLn="118" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="1" locEndLn="118" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="1" locEndLn="118" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="2" locEndLn="118" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="2" locEndLn="118" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="118" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="2" locEndLn="118" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_mutex\_lock' is called without scope resolution operator '::'" ln="118" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="118" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="118" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="118" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="118" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="118" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="118" locStartPos="3" locEndLn="118" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<FlowViol msg="&quot;ring&quot; is used in two critical sections in context of single method, using one critical section will improve atomicity of operation" ln="119" ruleSAFMsg="Usage of &quot;ring&quot; in second critical section" auth="devtest" sev="2" rule="BD-TRS-DIFCS" ruleSCSCMsg="Usage of &quot;ring&quot; in first critical section" pkg="Render" tool="c++test" id="240447899" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="0" locEndLn="120" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="119" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="120" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props>

<Prop key="Tracked variables" val="Variable used in critical section"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="117" srcRngStartPos="0" srcRngEndLn="118" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="117" ElType="." desc="while (!exitGame) {" rngLn="117">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="118" srcRngStartPos="0" srcRngEndLn="119" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="118" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="118">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="119" srcRngStartPos="0" srcRngEndLn="120" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="119" ElType=".C" desc="ring.getPosition()" rngLn="119">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Usage of &quot;ring&quot; in first critical section" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="119" srcRngStartPos="0" srcRngEndLn="120" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="119" ElType="." desc="ring.getPosition().translate(\*currentCameraVelocity);" rngLn="119">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="120" srcRngStartPos="0" srcRngEndLn="121" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="120" ElType="!" desc="LOCK\_RELEASE(changePositionMutex);" rngLn="120">

<Props/>

<Anns>

<Ann msg="Unlocking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="122" srcRngStartPos="0" srcRngEndLn="123" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="122" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="122">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming false)" kind="condEval"/>

<Ann msg="Not entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="128" srcRngStartPos="0" srcRngEndLn="129" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="128" ElType="." desc="SLEEP(STEP);" rngLn="128">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="117" srcRngStartPos="0" srcRngEndLn="118" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="117" ElType="." desc="while (!exitGame) {" rngLn="117">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="118" srcRngStartPos="0" srcRngEndLn="119" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="118" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="118">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="119" srcRngStartPos="0" srcRngEndLn="120" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="119" ElType=".P" desc="ring.getPosition()" rngLn="119">

<Props/>

<Anns>

<Ann msg="Usage of &quot;ring&quot; in second critical section" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Non-ascii tab found" ln="119" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="0" locEndLn="119" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="119" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="0" locEndLn="119" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="119" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="0" locEndLn="119" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="119" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="1" locEndLn="119" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="119" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="1" locEndLn="119" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="119" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="1" locEndLn="119" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="119" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="2" locEndLn="119" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="119" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="2" locEndLn="119" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="119" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="119" locStartPos="2" locEndLn="119" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="0" locEndLn="120" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="0" locEndLn="120" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="0" locEndLn="120" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="1" locEndLn="120" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="1" locEndLn="120" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="1" locEndLn="120" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="2" locEndLn="120" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="2" locEndLn="120" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="120" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="2" locEndLn="120" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_mutex\_unlock' is called without scope resolution operator '::'" ln="120" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="120" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="120" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="120" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="120" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="120" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="120" locStartPos="3" locEndLn="120" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'i' shall be declared as unsigned int or signed int" ln="122" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="11" locEndLn="122" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="A loop does not have a fixed upper nor lower bound" ln="122" sev="3" auth="devtest" rule="CODSTA-83" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="3" locEndLn="122" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Between conditional statement 'for' and its opening parenthesis '(' should be exactly one space" ln="122" sev="3" auth="devtest" rule="FORMAT-12" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="3" locEndLn="122" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="1" locEndLn="122" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="1" locEndLn="122" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="1" locEndLn="122" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="2" locEndLn="122" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="2" locEndLn="122" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="122" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="2" locEndLn="122" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="122" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="122" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Postfix operator applied to variable 'i'; prefer prefix type" ln="122" sev="3" auth="devtest" rule="OPT-04" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="41" locEndLn="122" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="122" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="122" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="122" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="122" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="0" locEndLn="122" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'i' variable should be commented" ln="122" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="11" locEndLn="122" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'i' variable should be commented" ln="122" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="11" locEndLn="122" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="122" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="122" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="122" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="122" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="122" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="122" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="122" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="122" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'i' variable should contain a braced initializer" ln="122" sev="2" auth="devtest" rule="AUTOSAR-A8\_5\_2-a" tool="c++test" cat="AUTOSAR-A8\_5\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="11" locEndLn="122" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'i' variable should contain a braced initializer" ln="122" sev="3" auth="devtest" rule="CODSTA-MCPP-38" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="11" locEndLn="122" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="122" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="122" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="7" locEndLn="122" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Variable &quot;i&quot; declaration was found in control statement" ln="122" sev="3" auth="devtest" rule="OPT-10" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="122" locStartPos="11" locEndLn="122" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="0" locEndLn="123" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="0" locEndLn="123" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="0" locEndLn="123" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="1" locEndLn="123" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="1" locEndLn="123" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="1" locEndLn="123" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="2" locEndLn="123" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="2" locEndLn="123" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="2" locEndLn="123" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="3" locEndLn="123" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="3" locEndLn="123" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="123" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="3" locEndLn="123" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_mutex\_lock' is called without scope resolution operator '::'" ln="123" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_lock&quot; return value" ln="123" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="123" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="123" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="123" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_lock' return value" ln="123" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="123" locStartPos="4" locEndLn="123" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<FlowViol msg="&quot;currentCameraVelocity&quot; is used in two critical sections in context of single method, using one critical section will improve atomicity of operation" ln="124" ruleSAFMsg="Usage of &quot;currentCameraVelocity&quot; in second critical section" auth="devtest" sev="2" rule="BD-TRS-DIFCS" ruleSCSCMsg="Usage of &quot;currentCameraVelocity&quot; in first critical section" pkg="Render" tool="c++test" id="-881739154" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="0" locEndLn="125" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="119" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="120" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props>

<Prop key="Tracked variables" val="Variable used in critical section"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="117" srcRngStartPos="0" srcRngEndLn="118" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="117" ElType="." desc="while (!exitGame) {" rngLn="117">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="118" srcRngStartPos="0" srcRngEndLn="119" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="118" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="118">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="119" srcRngStartPos="0" srcRngEndLn="120" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="119" ElType="." desc="ring.getPosition()" rngLn="119">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="119" srcRngStartPos="0" srcRngEndLn="120" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="119" ElType=".C" desc="ring.getPosition().translate(\*currentCameraVelocity);" rngLn="119">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Usage of &quot;currentCameraVelocity&quot; in first critical section" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="120" srcRngStartPos="0" srcRngEndLn="121" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="120" ElType="!" desc="LOCK\_RELEASE(changePositionMutex);" rngLn="120">

<Props/>

<Anns>

<Ann msg="Unlocking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="122" srcRngStartPos="0" srcRngEndLn="123" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="122" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="122">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="123" srcRngStartPos="0" srcRngEndLn="124" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="123" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="123">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="124" srcRngStartPos="0" srcRngEndLn="125" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="124" ElType="." desc="participants[i]->getPosition()" rngLn="124">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="124" srcRngStartPos="0" srcRngEndLn="125" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="124" ElType=".P" desc="participants[i]->getPosition().translate(\*currentCameraVelocity);" rngLn="124">

<Props/>

<Anns>

<Ann msg="Usage of &quot;currentCameraVelocity&quot; in second critical section" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;participants&quot; is used in two critical sections in context of single method, using one critical section will improve atomicity of operation" ln="124" ruleSAFMsg="Usage of &quot;participants&quot; in second critical section" auth="devtest" sev="2" rule="BD-TRS-DIFCS" ruleSCSCMsg="Usage of &quot;participants&quot; in first critical section" pkg="Render" tool="c++test" id="-935999178" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="0" locEndLn="125" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" FirstElSrcRngStartln="124" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="125" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<Props>

<Prop key="Tracked variables" val="Variable used in critical section"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="117" srcRngStartPos="0" srcRngEndLn="118" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="117" ElType="." desc="while (!exitGame) {" rngLn="117">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: !exitGame (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="118" srcRngStartPos="0" srcRngEndLn="119" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="118" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="118">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="119" srcRngStartPos="0" srcRngEndLn="120" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="119" ElType="." desc="ring.getPosition()" rngLn="119">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="119" srcRngStartPos="0" srcRngEndLn="120" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="119" ElType="." desc="ring.getPosition().translate(\*currentCameraVelocity);" rngLn="119">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="120" srcRngStartPos="0" srcRngEndLn="121" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="120" ElType="!" desc="LOCK\_RELEASE(changePositionMutex);" rngLn="120">

<Props/>

<Anns>

<Ann msg="Unlocking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="122" srcRngStartPos="0" srcRngEndLn="123" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="122" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="122">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="123" srcRngStartPos="0" srcRngEndLn="124" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="123" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="123">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="124" srcRngStartPos="0" srcRngEndLn="125" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="124" ElType=".C" desc="participants[i]->getPosition()" rngLn="124">

<ElDescList>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="10" ElType="." desc="Point&amp; getPosition() { return \_position; }" rngLn="10">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Usage of &quot;participants&quot; in first critical section" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="124" srcRngStartPos="0" srcRngEndLn="125" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="124" ElType="." desc="participants[i]->getPosition().translate(\*currentCameraVelocity);" rngLn="124">

<ElDescList>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="15" ElType="." desc="\_x += vector.\_x;" rngLn="15">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="16" srcRngStartPos="0" srcRngEndLn="17" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" srcRnghash="1950870755" ln="16" ElType="." desc="\_y += vector.\_y;" rngLn="16">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

</ElDesc>

<ElDesc srcRngStartln="125" srcRngStartPos="0" srcRngEndLn="126" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="125" ElType="!" desc="LOCK\_RELEASE(changePositionMutex);" rngLn="125">

<Props/>

<Anns>

<Ann msg="Unlocking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="126" srcRngStartPos="0" srcRngEndLn="127" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="126" ElType="!" desc="draw(participants[i]);" rngLn="126">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="122" srcRngStartPos="0" srcRngEndLn="123" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="122" ElType="." desc="for(int i = 0; i &lt; participantsCount; i++) {" rngLn="122">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; participantsCount) (true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="123" srcRngStartPos="0" srcRngEndLn="124" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="123" ElType="!" desc="LOCK\_ACQUIRE(changePositionMutex);" rngLn="123">

<Props/>

<Anns>

<Ann msg="Locking: &amp;changePositionMutex" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="124" srcRngStartPos="0" srcRngEndLn="125" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="124" ElType=".P" desc="participants[i]->getPosition()" rngLn="124">

<Props/>

<Anns>

<Ann msg="Usage of &quot;participants&quot; in second critical section" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Non-ascii tab found" ln="124" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="0" locEndLn="124" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="0" locEndLn="124" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="0" locEndLn="124" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="1" locEndLn="124" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="1" locEndLn="124" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="1" locEndLn="124" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="2" locEndLn="124" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="2" locEndLn="124" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="2" locEndLn="124" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="3" locEndLn="124" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="3" locEndLn="124" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="124" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="124" locStartPos="3" locEndLn="124" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="0" locEndLn="125" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="0" locEndLn="125" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="0" locEndLn="125" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="1" locEndLn="125" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="1" locEndLn="125" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="1" locEndLn="125" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="2" locEndLn="125" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="2" locEndLn="125" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="2" locEndLn="125" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="3" locEndLn="125" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="3" locEndLn="125" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="125" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="3" locEndLn="125" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_mutex\_unlock' is called without scope resolution operator '::'" ln="125" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_mutex\_unlock&quot; return value" ln="125" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="125" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="125" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="125" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_mutex\_unlock' return value" ln="125" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="125" locStartPos="4" locEndLn="125" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="0" locEndLn="126" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="0" locEndLn="126" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="0" locEndLn="126" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="1" locEndLn="126" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="1" locEndLn="126" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="1" locEndLn="126" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="2" locEndLn="126" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="2" locEndLn="126" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="2" locEndLn="126" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="3" locEndLn="126" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="3" locEndLn="126" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="126" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="126" locStartPos="3" locEndLn="126" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="0" locEndLn="127" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="0" locEndLn="127" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="0" locEndLn="127" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="1" locEndLn="127" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="1" locEndLn="127" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="1" locEndLn="127" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="2" locEndLn="127" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="2" locEndLn="127" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="127" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="127" locStartPos="2" locEndLn="127" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="A 'U' suffix shall be applied to constant: 20" ln="128" sev="3" auth="devtest" rule="HICPP-4\_2\_1-a" tool="c++test" cat="HICPP-4\_2\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="9" locEndLn="128" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion between signed and unsigned type in the function call on argument '1' shall not be used" ln="128" sev="2" auth="devtest" rule="MISRA2008-5\_0\_4\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="9" locEndLn="128" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion between signed and unsigned type in the function call on argument '1' shall not be used" ln="128" sev="2" auth="devtest" rule="AUTOSAR-M5\_0\_4-a" tool="c++test" cat="AUTOSAR-M5\_0\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="9" locEndLn="128" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Implicit conversion between signed and unsigned type in the function call on argument '1' shall not be used" ln="128" sev="3" auth="devtest" rule="MISRA2004-10\_1\_a" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="9" locEndLn="128" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="0" locEndLn="128" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="0" locEndLn="128" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="0" locEndLn="128" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="1" locEndLn="128" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="1" locEndLn="128" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="1" locEndLn="128" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="2" locEndLn="128" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="2" locEndLn="128" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="128" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="2" locEndLn="128" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'sleep' is called without scope resolution operator '::'" ln="128" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The type 'int' of function argument number '1' does not match declared type 'unsigned int'" ln="128" sev="3" auth="devtest" rule="PB-11" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;sleep&quot; return value" ln="128" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'sleep' return value" ln="128" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'sleep' return value" ln="128" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'sleep' return value" ln="128" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'sleep' return value" ln="128" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="128" locStartPos="3" locEndLn="128" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="129" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="129" locStartPos="0" locEndLn="129" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="129" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="129" locStartPos="0" locEndLn="129" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="129" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="129" locStartPos="0" locEndLn="129" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="129" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="129" locStartPos="1" locEndLn="129" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="129" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="129" locStartPos="1" locEndLn="129" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="129" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="129" locStartPos="1" locEndLn="129" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="130" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="2" locEndLn="130" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="130" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="130" sev="2" auth="devtest" rule="AUTOSAR-M4\_10\_2-a" tool="c++test" cat="AUTOSAR-M4\_10\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="130" sev="2" auth="devtest" rule="MISRA2008-4\_10\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="130" sev="3" auth="devtest" rule="CODSTA-CPP-63" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="130" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-a" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="130" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="130" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function with pointer return type returns '0'" ln="130" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="130" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="0" locEndLn="130" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="130" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="0" locEndLn="130" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="130" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="0" locEndLn="130" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="130" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="1" locEndLn="130" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="130" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="1" locEndLn="130" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="130" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="1" locEndLn="130" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="130" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="130" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="130" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="130" locStartPos="9" locEndLn="130" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="131" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="131" locStartPos="0" locEndLn="131" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="131" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="131" locStartPos="0" locEndLn="131" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="131" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="131" locStartPos="0" locEndLn="131" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has Cyclomatic Complexity value: 1" ln="134" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has Essential Complexity value: 1" ln="134" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has external linkage and is not declared in the header" ln="134" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has external linkage and is not declared in the header" ln="134" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has external linkage and is not declared in the header" ln="134" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has external linkage and is not declared in the header" ln="134" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has external linkage and is not declared in the header" ln="134" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has external linkage and is not declared in the header" ln="134" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads' has external linkage and is not declared in the header" ln="134" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'runGameThreads' is declared in global namespace" ln="134" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'runGameThreads' is declared in global namespace" ln="134" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'runGameThreads' is declared in global namespace" ln="134" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'runGameThreads' is declared in global namespace" ln="134" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: runGameThreads" ln="134" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Return type is not placed in line before function 'runGameThreads'" ln="134" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads' function is not used in the testing scope" ln="134" sev="3" auth="devtest" rule="GLOBAL-UNUSEDFUNC" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads' function is not used in the testing scope" ln="134" sev="4" auth="devtest" rule="AUTOSAR-M0\_1\_10-a" tool="c++test" cat="AUTOSAR-M0\_1\_10" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads' function is not used in the testing scope" ln="134" sev="2" auth="devtest" rule="MISRA2008-0\_1\_10\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads' function should be preceded by a comment that contains the '@brief' tag" ln="134" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads' function should be preceded by a comment that contains the '@brief' tag" ln="134" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'runGameThreads' function is not preceded by a comment" ln="134" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'runGameThreads' function is not preceded by a comment" ln="134" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'runGameThreads' should be composed only of lowercase letters" ln="134" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'runGameThreads' should be composed only of lowercase letters" ln="134" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="134" locStartPos="5" locEndLn="134" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<DupViol msg="Duplicated function: 'void runGameThreads ( ) { exitGame = 0 ; THREAD thread1 ,...'" ln="135" NvType="1" sev="2" auth="devtest" rule="CDD-DUPM" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="-1769734618" NvActs="3" locStartln="135" locStartPos="0" locEndLn="142" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<ElDescList>

<ElDesc srcRngStartln="135" srcRngStartPos="0" srcRngEndLn="142" srcRngEndPos="1" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="135" ElType="" desc="[Line 135] Duplicated function in file 'DeadLock.cpp'" sourceRngHash="1072561057">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="145" srcRngStartPos="0" srcRngEndLn="152" srcRngEndPos="1" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="145" ElType="" desc="[Line 145] Duplicated function in file 'DeadLock.cpp'" sourceRngHash="1072561057">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="135" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="135" locStartPos="0" locEndLn="135" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<DupViol msg="Duplicated code: 'exitGame = 0;&#x9;THREAD thread1, thread2, thread3, thread4;&#x9;THR...'" ln="136" NvType="1" sev="3" auth="devtest" rule="CDD-DUPC" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="-1769734618" NvActs="3" locStartln="136" locStartPos="1" locEndLn="141" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp">

<ElDescList>

<ElDesc srcRngStartln="136" srcRngStartPos="1" srcRngEndLn="141" srcRngEndPos="47" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="136" ElType="" desc="[Line 136] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="-348868364">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="146" srcRngStartPos="1" srcRngEndLn="151" srcRngEndPos="47" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" srcRnghash="-1769734618" ln="146" ElType="" desc="[Line 146] Duplicated code in file 'DeadLock.cpp'" sourceRngHash="-348868364">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Global variable 'exitGame' is modified in function 'runGameThreads'" ln="136" sev="3" auth="devtest" rule="CODSTA-27" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="136" locStartPos="1" locEndLn="136" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="136" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="136" locStartPos="0" locEndLn="136" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="136" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="136" locStartPos="0" locEndLn="136" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="136" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="136" locStartPos="0" locEndLn="136" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-c" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="FORMAT-33" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="MISRA2008-8\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="HICPP-7\_1\_1-b" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="AUTOSAR-M8\_0\_1-a" tool="c++test" cat="AUTOSAR-M8\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="CERT\_C-DCL04-a" tool="c++test" cat="CERT\_C-DCL04" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate line" ln="137" sev="2" auth="devtest" rule="JSF-152" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate line" ln="137" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-b" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate line" ln="137" sev="3" auth="devtest" rule="HICPP-7\_1\_1-a" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate line" ln="137" sev="3" auth="devtest" rule="FORMAT-29" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-c" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="FORMAT-33" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="MISRA2008-8\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="HICPP-7\_1\_1-b" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="AUTOSAR-M8\_0\_1-a" tool="c++test" cat="AUTOSAR-M8\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="CERT\_C-DCL04-a" tool="c++test" cat="CERT\_C-DCL04" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate line" ln="137" sev="2" auth="devtest" rule="JSF-152" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate line" ln="137" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-b" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate line" ln="137" sev="3" auth="devtest" rule="HICPP-7\_1\_1-a" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate line" ln="137" sev="3" auth="devtest" rule="FORMAT-29" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-c" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="FORMAT-33" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="MISRA2008-8\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="HICPP-7\_1\_1-b" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="137" sev="2" auth="devtest" rule="AUTOSAR-M8\_0\_1-a" tool="c++test" cat="AUTOSAR-M8\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="137" sev="3" auth="devtest" rule="CERT\_C-DCL04-a" tool="c++test" cat="CERT\_C-DCL04" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate line" ln="137" sev="2" auth="devtest" rule="JSF-152" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate line" ln="137" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-b" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate line" ln="137" sev="3" auth="devtest" rule="HICPP-7\_1\_1-a" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate line" ln="137" sev="3" auth="devtest" rule="FORMAT-29" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="137" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="0" locEndLn="137" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="137" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="0" locEndLn="137" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="137" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="0" locEndLn="137" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'thread1' uninitialized when declared" ln="137" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="11" locEndLn="137" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'thread2' uninitialized when declared" ln="137" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'thread3' uninitialized when declared" ln="137" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'thread4' uninitialized when declared" ln="137" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable has only one use" ln="137" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="11" locEndLn="137" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable has only one use" ln="137" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="11" locEndLn="137" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable has only one use" ln="137" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="11" locEndLn="137" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable should be commented" ln="137" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="11" locEndLn="137" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable should be commented" ln="137" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="11" locEndLn="137" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable has only one use" ln="137" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable has only one use" ln="137" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable has only one use" ln="137" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable should be commented" ln="137" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable should be commented" ln="137" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable has only one use" ln="137" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable has only one use" ln="137" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable has only one use" ln="137" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable should be commented" ln="137" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable should be commented" ln="137" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable has only one use" ln="137" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable has only one use" ln="137" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable has only one use" ln="137" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable should be commented" ln="137" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable should be commented" ln="137" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: thread1" ln="137" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="11" locEndLn="137" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: thread2" ln="137" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: thread3" ln="137" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: thread4" ln="137" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: thread1" ln="137" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="11" locEndLn="137" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: thread2" ln="137" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="20" locEndLn="137" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: thread3" ln="137" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="29" locEndLn="137" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: thread4" ln="137" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="137" locStartPos="38" locEndLn="137" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="138" sev="3" auth="devtest" rule="CERT\_CPP-EXP56-a" tool="c++test" cat="CERT\_CPP-EXP56" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="29" locEndLn="138" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="138" sev="3" auth="devtest" rule="CODSTA-CPP-96" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="29" locEndLn="138" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="138" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="0" locEndLn="138" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="138" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="0" locEndLn="138" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="138" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="0" locEndLn="138" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="138" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="26" locEndLn="138" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="138" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="26" locEndLn="138" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="138" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="26" locEndLn="138" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="138" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="41" locEndLn="138" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="138" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="41" locEndLn="138" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="138" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="41" locEndLn="138" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_create' is called without scope resolution operator '::'" ln="138" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="138" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="26" locEndLn="138" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="138" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="26" locEndLn="138" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="138" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="26" locEndLn="138" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="138" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="41" locEndLn="138" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="138" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="41" locEndLn="138" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="138" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="41" locEndLn="138" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="138" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="138" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="138" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="138" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="138" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="138" locStartPos="1" locEndLn="138" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="139" sev="3" auth="devtest" rule="CERT\_CPP-EXP56-a" tool="c++test" cat="CERT\_CPP-EXP56" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="29" locEndLn="139" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="139" sev="3" auth="devtest" rule="CODSTA-CPP-96" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="29" locEndLn="139" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="139" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="0" locEndLn="139" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="139" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="0" locEndLn="139" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="139" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="0" locEndLn="139" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="139" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="26" locEndLn="139" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="139" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="26" locEndLn="139" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="139" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="26" locEndLn="139" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="139" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="42" locEndLn="139" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="139" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="42" locEndLn="139" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="139" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="42" locEndLn="139" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_create' is called without scope resolution operator '::'" ln="139" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="139" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="26" locEndLn="139" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="139" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="26" locEndLn="139" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="139" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="26" locEndLn="139" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="139" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="42" locEndLn="139" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="139" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="42" locEndLn="139" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="139" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="42" locEndLn="139" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="139" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="139" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="139" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="139" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="139" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="139" locStartPos="1" locEndLn="139" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="140" sev="3" auth="devtest" rule="CERT\_CPP-EXP56-a" tool="c++test" cat="CERT\_CPP-EXP56" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="29" locEndLn="140" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="140" sev="3" auth="devtest" rule="CODSTA-CPP-96" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="29" locEndLn="140" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="140" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="0" locEndLn="140" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="140" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="0" locEndLn="140" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="140" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="0" locEndLn="140" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="140" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="26" locEndLn="140" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="140" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="26" locEndLn="140" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="140" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="26" locEndLn="140" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="140" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="48" locEndLn="140" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="140" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="48" locEndLn="140" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="140" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="48" locEndLn="140" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_create' is called without scope resolution operator '::'" ln="140" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="140" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="26" locEndLn="140" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="140" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="26" locEndLn="140" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="140" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="26" locEndLn="140" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="140" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="48" locEndLn="140" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="140" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="48" locEndLn="140" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="140" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="48" locEndLn="140" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="140" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="140" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="140" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="140" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="140" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="140" locStartPos="1" locEndLn="140" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="141" sev="3" auth="devtest" rule="CERT\_CPP-EXP56-a" tool="c++test" cat="CERT\_CPP-EXP56" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="29" locEndLn="141" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="141" sev="3" auth="devtest" rule="CODSTA-CPP-96" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="29" locEndLn="141" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="141" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="0" locEndLn="141" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="141" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="0" locEndLn="141" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="141" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="0" locEndLn="141" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="141" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="26" locEndLn="141" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="141" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="26" locEndLn="141" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="141" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="26" locEndLn="141" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="141" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="46" locEndLn="141" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="141" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="46" locEndLn="141" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="141" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="46" locEndLn="141" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_create' is called without scope resolution operator '::'" ln="141" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="141" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="26" locEndLn="141" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="141" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="26" locEndLn="141" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="141" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="26" locEndLn="141" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="141" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="46" locEndLn="141" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="141" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="46" locEndLn="141" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="141" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="46" locEndLn="141" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="141" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="141" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="141" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="141" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="141" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="141" locStartPos="1" locEndLn="141" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has Cyclomatic Complexity value: 1" ln="144" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has Essential Complexity value: 1" ln="144" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has external linkage and is not declared in the header" ln="144" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has external linkage and is not declared in the header" ln="144" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has external linkage and is not declared in the header" ln="144" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has external linkage and is not declared in the header" ln="144" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has external linkage and is not declared in the header" ln="144" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has external linkage and is not declared in the header" ln="144" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Function 'runGameThreads2' has external linkage and is not declared in the header" ln="144" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'runGameThreads2' is declared in global namespace" ln="144" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'runGameThreads2' is declared in global namespace" ln="144" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'runGameThreads2' is declared in global namespace" ln="144" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global function 'runGameThreads2' is declared in global namespace" ln="144" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Naming convention not followed: runGameThreads2" ln="144" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Return type is not placed in line before function 'runGameThreads2'" ln="144" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads2' function is not used in the testing scope" ln="144" sev="3" auth="devtest" rule="GLOBAL-UNUSEDFUNC" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads2' function is not used in the testing scope" ln="144" sev="4" auth="devtest" rule="AUTOSAR-M0\_1\_10-a" tool="c++test" cat="AUTOSAR-M0\_1\_10" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads2' function is not used in the testing scope" ln="144" sev="2" auth="devtest" rule="MISRA2008-0\_1\_10\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads2' function should be preceded by a comment that contains the '@brief' tag" ln="144" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'runGameThreads2' function should be preceded by a comment that contains the '@brief' tag" ln="144" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'runGameThreads2' function is not preceded by a comment" ln="144" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The definition of the 'runGameThreads2' function is not preceded by a comment" ln="144" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'runGameThreads2' should be composed only of lowercase letters" ln="144" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The name 'runGameThreads2' should be composed only of lowercase letters" ln="144" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="144" locStartPos="5" locEndLn="144" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="145" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="145" locStartPos="0" locEndLn="145" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Global variable 'exitGame' is modified in function 'runGameThreads2'" ln="146" sev="3" auth="devtest" rule="CODSTA-27" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="146" locStartPos="1" locEndLn="146" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="146" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="146" locStartPos="0" locEndLn="146" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="146" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="146" locStartPos="0" locEndLn="146" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="146" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="146" locStartPos="0" locEndLn="146" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-c" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="FORMAT-33" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="MISRA2008-8\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="HICPP-7\_1\_1-b" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="AUTOSAR-M8\_0\_1-a" tool="c++test" cat="AUTOSAR-M8\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="CERT\_C-DCL04-a" tool="c++test" cat="CERT\_C-DCL04" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate line" ln="147" sev="2" auth="devtest" rule="JSF-152" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate line" ln="147" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-b" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate line" ln="147" sev="3" auth="devtest" rule="HICPP-7\_1\_1-a" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread2' in a separate line" ln="147" sev="3" auth="devtest" rule="FORMAT-29" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-c" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="FORMAT-33" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="MISRA2008-8\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="HICPP-7\_1\_1-b" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="AUTOSAR-M8\_0\_1-a" tool="c++test" cat="AUTOSAR-M8\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="CERT\_C-DCL04-a" tool="c++test" cat="CERT\_C-DCL04" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate line" ln="147" sev="2" auth="devtest" rule="JSF-152" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate line" ln="147" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-b" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate line" ln="147" sev="3" auth="devtest" rule="HICPP-7\_1\_1-a" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread3' in a separate line" ln="147" sev="3" auth="devtest" rule="FORMAT-29" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-c" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="FORMAT-33" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="MISRA2008-8\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="HICPP-7\_1\_1-b" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="147" sev="2" auth="devtest" rule="AUTOSAR-M8\_0\_1-a" tool="c++test" cat="AUTOSAR-M8\_0\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate declaration statement" ln="147" sev="3" auth="devtest" rule="CERT\_C-DCL04-a" tool="c++test" cat="CERT\_C-DCL04" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate line" ln="147" sev="2" auth="devtest" rule="JSF-152" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate line" ln="147" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-b" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate line" ln="147" sev="3" auth="devtest" rule="HICPP-7\_1\_1-a" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Declare variable 'thread4' in a separate line" ln="147" sev="3" auth="devtest" rule="FORMAT-29" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="147" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="0" locEndLn="147" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="147" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="0" locEndLn="147" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="147" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="0" locEndLn="147" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'thread1' uninitialized when declared" ln="147" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="11" locEndLn="147" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'thread2' uninitialized when declared" ln="147" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'thread3' uninitialized when declared" ln="147" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Pointer variable 'thread4' uninitialized when declared" ln="147" sev="2" auth="devtest" rule="INIT-04" tool="c++test" cat="INIT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable has only one use" ln="147" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="11" locEndLn="147" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable has only one use" ln="147" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="11" locEndLn="147" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable has only one use" ln="147" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="11" locEndLn="147" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable should be commented" ln="147" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="11" locEndLn="147" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread1' variable should be commented" ln="147" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="11" locEndLn="147" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable has only one use" ln="147" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable has only one use" ln="147" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable has only one use" ln="147" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable should be commented" ln="147" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread2' variable should be commented" ln="147" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable has only one use" ln="147" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable has only one use" ln="147" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable has only one use" ln="147" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable should be commented" ln="147" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread3' variable should be commented" ln="147" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable has only one use" ln="147" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_4-a" tool="c++test" cat="AUTOSAR-M0\_1\_4" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable has only one use" ln="147" sev="2" auth="devtest" rule="MISRA2008-0\_1\_4" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable has only one use" ln="147" sev="3" auth="devtest" rule="GLOBAL-ONEUSEVAR" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable should be commented" ln="147" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The 'thread4' variable should be commented" ln="147" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: thread1" ln="147" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="11" locEndLn="147" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: thread2" ln="147" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: thread3" ln="147" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: thread4" ln="147" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: thread1" ln="147" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="11" locEndLn="147" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: thread2" ln="147" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="20" locEndLn="147" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: thread3" ln="147" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="29" locEndLn="147" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The variable of pointer type is declared: thread4" ln="147" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="147" locStartPos="38" locEndLn="147" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="148" sev="3" auth="devtest" rule="CERT\_CPP-EXP56-a" tool="c++test" cat="CERT\_CPP-EXP56" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="29" locEndLn="148" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="148" sev="3" auth="devtest" rule="CODSTA-CPP-96" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="29" locEndLn="148" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="148" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="0" locEndLn="148" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="148" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="0" locEndLn="148" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="148" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="0" locEndLn="148" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="148" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="26" locEndLn="148" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="148" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="26" locEndLn="148" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="148" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="26" locEndLn="148" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="148" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="41" locEndLn="148" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="148" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="41" locEndLn="148" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="148" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="41" locEndLn="148" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_create' is called without scope resolution operator '::'" ln="148" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="148" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="26" locEndLn="148" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="148" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="26" locEndLn="148" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="148" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="26" locEndLn="148" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="148" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="41" locEndLn="148" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="148" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="41" locEndLn="148" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="148" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="41" locEndLn="148" locEndPos="42" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="148" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="148" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="148" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="148" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="148" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="148" locStartPos="1" locEndLn="148" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="149" sev="3" auth="devtest" rule="CERT\_CPP-EXP56-a" tool="c++test" cat="CERT\_CPP-EXP56" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="29" locEndLn="149" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="149" sev="3" auth="devtest" rule="CODSTA-CPP-96" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="29" locEndLn="149" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="149" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="0" locEndLn="149" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="149" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="0" locEndLn="149" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="149" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="0" locEndLn="149" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="149" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="26" locEndLn="149" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="149" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="26" locEndLn="149" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="149" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="26" locEndLn="149" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="149" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="42" locEndLn="149" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="149" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="42" locEndLn="149" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="149" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="42" locEndLn="149" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_create' is called without scope resolution operator '::'" ln="149" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="149" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="26" locEndLn="149" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="149" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="26" locEndLn="149" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="149" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="26" locEndLn="149" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="149" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="42" locEndLn="149" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="149" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="42" locEndLn="149" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="149" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="42" locEndLn="149" locEndPos="43" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="149" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="149" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="149" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="149" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="149" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="149" locStartPos="1" locEndLn="149" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="150" sev="3" auth="devtest" rule="CERT\_CPP-EXP56-a" tool="c++test" cat="CERT\_CPP-EXP56" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="29" locEndLn="150" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="150" sev="3" auth="devtest" rule="CODSTA-CPP-96" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="29" locEndLn="150" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="150" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="0" locEndLn="150" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="150" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="0" locEndLn="150" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="150" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="0" locEndLn="150" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="150" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="26" locEndLn="150" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="150" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="26" locEndLn="150" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="150" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="26" locEndLn="150" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="150" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="48" locEndLn="150" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="150" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="48" locEndLn="150" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="150" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="48" locEndLn="150" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_create' is called without scope resolution operator '::'" ln="150" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="150" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="26" locEndLn="150" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="150" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="26" locEndLn="150" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="150" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="26" locEndLn="150" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="150" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="48" locEndLn="150" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="150" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="48" locEndLn="150" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="150" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="48" locEndLn="150" locEndPos="49" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="150" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="150" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="150" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="150" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="150" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="150" locStartPos="1" locEndLn="150" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="151" sev="3" auth="devtest" rule="CERT\_CPP-EXP56-a" tool="c++test" cat="CERT\_CPP-EXP56" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="29" locEndLn="151" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Do not assing the pointer to the function with C++ language linkage to the pointer to the function with C language linkage" ln="151" sev="3" auth="devtest" rule="CODSTA-CPP-96" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="29" locEndLn="151" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="151" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="0" locEndLn="151" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="151" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="0" locEndLn="151" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Non-ascii tab found" ln="151" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="0" locEndLn="151" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="151" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="26" locEndLn="151" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="151" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="26" locEndLn="151" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="151" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="26" locEndLn="151" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="151" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="46" locEndLn="151" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="151" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="46" locEndLn="151" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="151" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="46" locEndLn="151" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The global function 'pthread\_create' is called without scope resolution operator '::'" ln="151" sev="5" auth="devtest" rule="CODSTA-CPP-23" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="151" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="26" locEndLn="151" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="151" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="26" locEndLn="151" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '2' argument to function 'pthread\_create' " ln="151" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="26" locEndLn="151" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="151" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="46" locEndLn="151" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="151" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="46" locEndLn="151" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="The value '0' is passed as '4' argument to function 'pthread\_create' " ln="151" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="46" locEndLn="151" locEndPos="47" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" urgent="true" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's &quot;pthread\_create&quot; return value" ln="151" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="151" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="151" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="151" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Unused function's 'pthread\_create' return value" ln="151" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1769734618" locStartln="151" locStartPos="1" locEndLn="151" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'DivisionByZero.cpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-02" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'DivisionByZero.cpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'DivisionByZero.cpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-03" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'DivisionByZero.cpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Disallowed #include notation is being used: &quot;Shapes.hpp&quot;" ln="1" sev="2" auth="devtest" rule="PREPROC-09" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Disallowed #include notation is being used: &quot;Shapes.hpp&quot;" ln="1" sev="2" auth="devtest" rule="JSF-033" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Implementation file 'DivisionByZero.cpp' should declare a local constant string that begins from characters &quot;@(#)&quot; " ln="1" sev="5" auth="devtest" rule="PFO-04" tool="c++test" cat="PFO" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Implementation file 'DivisionByZero.cpp' should have the file name extension &quot;.cc&quot;" ln="1" sev="3" auth="devtest" rule="NAMING-38" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Not a proper header file (\*.h ) is being included: &quot;Shapes.hpp&quot;" ln="1" sev="3" auth="devtest" rule="JSF-032" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Not a proper header file (\*.h ) is being included: &quot;Shapes.hpp&quot;" ln="1" sev="3" auth="devtest" rule="PREPROC-08" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The assertion density is lower than two assertions per function" ln="1" sev="3" auth="devtest" rule="METRICS-31" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The filename 'DivisionByZero.cpp' should be in lowercase" ln="1" sev="3" auth="devtest" rule="NAMING-03" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape1' as const" ln="3" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape1' as const" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape1' as const" ln="3" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape1' as const" ln="3" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape1' as const" ln="3" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape2' as const" ln="3" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape2' as const" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape2' as const" ln="3" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape2' as const" ln="3" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Declare parameter 'shape2' as const" ln="3" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has Cyclomatic Complexity value: 1" ln="3" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has Essential Complexity value: 1" ln="3" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has external linkage and is not declared in the header" ln="3" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has external linkage and is not declared in the header" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has external linkage and is not declared in the header" ln="3" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has external linkage and is not declared in the header" ln="3" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has external linkage and is not declared in the header" ln="3" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has external linkage and is not declared in the header" ln="3" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Function 'getShapeRatio' has external linkage and is not declared in the header" ln="3" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Global function 'getShapeRatio' is declared in global namespace" ln="3" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Global function 'getShapeRatio' is declared in global namespace" ln="3" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Global function 'getShapeRatio' is declared in global namespace" ln="3" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Global function 'getShapeRatio' is declared in global namespace" ln="3" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Naming convention not followed: getShapeRatio" ln="3" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Parameter 'shape1' is not validated before use" ln="3" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Parameter 'shape1' is not validated before use" ln="3" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Parameter 'shape2' is not validated before use" ln="3" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Parameter 'shape2' is not validated before use" ln="3" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape1&quot; with const specifier" ln="3" sev="2" auth="devtest" rule="AUTOSAR-M7\_1\_2-b" tool="c++test" cat="AUTOSAR-M7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape1&quot; with const specifier" ln="3" sev="4" auth="devtest" rule="MISRAC2012-RULE\_8\_13-a" tool="c++test" cat="MISRAC2012-RULE\_8\_13" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape1&quot; with const specifier" ln="3" sev="3" auth="devtest" rule="CERT\_C-DCL13-a" tool="c++test" cat="CERT\_C-DCL13" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape1&quot; with const specifier" ln="3" sev="3" auth="devtest" rule="MISRA2004-16\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape1&quot; with const specifier" ln="3" sev="4" auth="devtest" rule="MISRA2012-RULE-8\_13\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape1&quot; with const specifier" ln="3" sev="4" auth="devtest" rule="JSF-118" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape1&quot; with const specifier" ln="3" sev="2" auth="devtest" rule="MISRA2008-7\_1\_2\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape2&quot; with const specifier" ln="3" sev="2" auth="devtest" rule="AUTOSAR-M7\_1\_2-b" tool="c++test" cat="AUTOSAR-M7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape2&quot; with const specifier" ln="3" sev="4" auth="devtest" rule="MISRAC2012-RULE\_8\_13-a" tool="c++test" cat="MISRAC2012-RULE\_8\_13" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape2&quot; with const specifier" ln="3" sev="3" auth="devtest" rule="CERT\_C-DCL13-a" tool="c++test" cat="CERT\_C-DCL13" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape2&quot; with const specifier" ln="3" sev="3" auth="devtest" rule="MISRA2004-16\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape2&quot; with const specifier" ln="3" sev="4" auth="devtest" rule="MISRA2012-RULE-8\_13\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape2&quot; with const specifier" ln="3" sev="4" auth="devtest" rule="JSF-118" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Pass parameter &quot;shape2&quot; with const specifier" ln="3" sev="2" auth="devtest" rule="MISRA2008-7\_1\_2\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Return type is not placed in line before function 'getShapeRatio'" ln="3" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<FlowViol msg="The &quot;shape1&quot; parameter of the &quot;getShapeRatio&quot; function cannot be NULL and should be passed by reference" ln="3" ruleSAFMsg="Function header" auth="devtest" sev="2" rule="AUTOSAR-A8\_4\_10-a" ruleSCSCMsg="Point where the parameter is dereferenced" tool="c++test" id="-938638073" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="4" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="3" srcRngStartPos="0" srcRngEndLn="4" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="3" ElType=".P" desc="double getShapeRatio(Shape\* shape1, Shape\* shape2)" rngLn="3">

<Props/>

<Anns>

<Ann msg="Function header" kind="point"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape1->getArea()" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point where the parameter is dereferenced" kind="cause"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="The &quot;shape1&quot; parameter of the &quot;getShapeRatio&quot; function cannot be NULL and should be passed by reference" ln="3" ruleSAFMsg="Function header" auth="devtest" sev="3" rule="BD-PB-REFPARAM" ruleSCSCMsg="Point where the parameter is dereferenced" tool="c++test" id="-1776184541" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="4" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="3" srcRngStartPos="0" srcRngEndLn="4" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="3" ElType=".P" desc="double getShapeRatio(Shape\* shape1, Shape\* shape2)" rngLn="3">

<Props/>

<Anns>

<Ann msg="Function header" kind="point"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape1->getArea()" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point where the parameter is dereferenced" kind="cause"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="The &quot;shape2&quot; parameter of the &quot;getShapeRatio&quot; function cannot be NULL and should be passed by reference" ln="3" ruleSAFMsg="Function header" auth="devtest" sev="2" rule="AUTOSAR-A8\_4\_10-a" ruleSCSCMsg="Point where the parameter is dereferenced" tool="c++test" id="1613810146" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="4" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="3" srcRngStartPos="0" srcRngEndLn="4" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="3" ElType=".P" desc="double getShapeRatio(Shape\* shape1, Shape\* shape2)" rngLn="3">

<Props/>

<Anns>

<Ann msg="Function header" kind="point"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point where the parameter is dereferenced" kind="cause"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="The &quot;shape2&quot; parameter of the &quot;getShapeRatio&quot; function cannot be NULL and should be passed by reference" ln="3" ruleSAFMsg="Function header" auth="devtest" sev="3" rule="BD-PB-REFPARAM" ruleSCSCMsg="Point where the parameter is dereferenced" tool="c++test" id="1042079686" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="4" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="3" srcRngStartPos="0" srcRngEndLn="4" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="3" ElType=".P" desc="double getShapeRatio(Shape\* shape1, Shape\* shape2)" rngLn="3">

<Props/>

<Anns>

<Ann msg="Function header" kind="point"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point where the parameter is dereferenced" kind="cause"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="The 'getShapeRatio' function is not used in the testing scope" ln="3" sev="3" auth="devtest" rule="GLOBAL-UNUSEDFUNC" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'getShapeRatio' function is not used in the testing scope" ln="3" sev="4" auth="devtest" rule="AUTOSAR-M0\_1\_10-a" tool="c++test" cat="AUTOSAR-M0\_1\_10" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'getShapeRatio' function is not used in the testing scope" ln="3" sev="2" auth="devtest" rule="MISRA2008-0\_1\_10\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'getShapeRatio' function should be preceded by a comment that contains the '@brief' tag" ln="3" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'getShapeRatio' function should be preceded by a comment that contains the '@brief' tag" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'getShapeRatio' function should be preceded by a comment that contains the '@return' tag" ln="3" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'getShapeRatio' function should be preceded by a comment that contains the '@return' tag" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'shape1' identifier should have the 'p' prefix" ln="3" sev="3" auth="devtest" rule="NAMING-HN-34" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'shape1' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="3" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'shape1' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'shape2' identifier should have the 'p' prefix" ln="3" sev="3" auth="devtest" rule="NAMING-HN-34" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'shape2' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="3" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The 'shape2' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="3" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="3" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="3" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="3" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="3" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="3" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="3" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="3" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The definition of the 'getShapeRatio' function is not preceded by a comment" ln="3" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The definition of the 'getShapeRatio' function is not preceded by a comment" ln="3" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The name 'getShapeRatio' should be composed only of lowercase letters" ln="3" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The name 'getShapeRatio' should be composed only of lowercase letters" ln="3" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared: shape1" ln="3" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared: shape2" ln="3" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The parameter of pointer type is declared: shape1" ln="3" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="28" locEndLn="3" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The parameter of pointer type is declared: shape2" ln="3" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="43" locEndLn="3" locEndPos="44" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="The return type of the 'getShapeRatio' function should be declared as 'auto'" ln="3" sev="2" auth="devtest" rule="CODSTA-MCPP-08\_b" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="3" locStartPos="7" locEndLn="3" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="4" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="4" locStartPos="0" locEndLn="4" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="5" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="1" locEndLn="5" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="2" rule="AUTOSAR-A5\_6\_1-a" ruleSCSCMsg="Start of the path" tool="c++test" id="-1887152637" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="2" rule="AUTOSAR-M0\_3\_1-e" ruleSCSCMsg="Start of the path" tool="c++test" id="-643759501" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="1" rule="BD-PB-ZERO" ruleSCSCMsg="Start of the path" tool="c++test" id="2097929330" lang="cpp" locType="sr" urgent="true" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="3" rule="CERT\_C-FLP03-a" ruleSCSCMsg="Start of the path" tool="c++test" id="-1966890730" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="2" rule="CERT\_C-INT33-a" ruleSCSCMsg="Start of the path" tool="c++test" id="-395738124" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="1" rule="HICPP-5\_5\_1-a" ruleSCSCMsg="Start of the path" tool="c++test" id="566822700" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="5" rule="MISRA2008-0\_3\_1\_c" ruleSCSCMsg="Start of the path" tool="c++test" id="-56152077" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="2" rule="MISRA2012-DIR-4\_1\_c" ruleSCSCMsg="Start of the path" tool="c++test" id="1579886536" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="2" rule="MISRA2012-RULE-1\_3\_a" ruleSCSCMsg="Start of the path" tool="c++test" id="-1579699948" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="2" rule="MISRAC2012-DIR\_4\_1-c" ruleSCSCMsg="Start of the path" tool="c++test" id="2109829747" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Division by &quot;shape2.getArea()&quot; which may possibly be zero" ln="5" ruleSAFMsg="Point of division by zero" auth="devtest" sev="2" rule="MISRAC2012-RULE\_1\_3-a" ruleSCSCMsg="Start of the path" tool="c++test" id="-1793694323" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="6" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" FirstElSrcRngStartln="5" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="6" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp">

<Props/>

<ElDescList>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".C" desc="shape2->getArea()" rngLn="5">

<ElDescList>

<ElDesc srcRngStartln="19" srcRngStartPos="0" srcRngEndLn="20" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" srcRnghash="1537905639" ln="19" ElType="!" desc="double getArea() const { return 0; }" rngLn="19">

<Props/>

</ElDesc>

</ElDescList>

<Props/>

<Anns>

<Ann msg="Start of the path" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="5" srcRngStartPos="0" srcRngEndLn="6" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" srcRnghash="-1665970746" ln="5" ElType=".P" desc="return shape1->getArea()/shape2->getArea();" rngLn="5">

<Props/>

<Anns>

<Ann msg="Point of division by zero" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Non-ascii tab found" ln="5" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Non-ascii tab found" ln="5" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="Non-ascii tab found" ln="5" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="5" sev="5" auth="devtest" rule="MISRA2008-0\_4\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="5" sev="3" auth="devtest" rule="OWASP2019-API9-d" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="5" sev="3" auth="devtest" rule="COMMENT-10" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="5" sev="2" auth="devtest" rule="AUTOSAR-M0\_4\_2-a" tool="c++test" cat="AUTOSAR-M0\_4\_2" lang="cpp" locType="sr" config="1" hash="-1665970746" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp"/>

<StdViol msg="&lt;stdio.h> library shall not be used" ln="1" sev="3" auth="devtest" rule="PREPROC-18" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="&lt;stdio.h> library shall not be used" ln="1" sev="2" auth="devtest" rule="JSF-022" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="&lt;stdio.h> library shall not be used" ln="1" sev="2" auth="devtest" rule="AUTOSAR-M27\_0\_1-a" tool="c++test" cat="AUTOSAR-M27\_0\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="&lt;stdio.h> library shall not be used" ln="1" sev="2" auth="devtest" rule="MISRA2008-27\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="&lt;stdio.h> library shall not be used" ln="1" sev="3" auth="devtest" rule="MISRA2004-20\_9" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'MemoryLeak.cpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-02" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'MemoryLeak.cpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'MemoryLeak.cpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-03" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'MemoryLeak.cpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Ensure that C Standard Library is securely used" ln="1" sev="4" auth="devtest" rule="HICPP-17\_2\_1-a" tool="c++test" cat="HICPP-17\_2\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Implementation file 'MemoryLeak.cpp' should declare a local constant string that begins from characters &quot;@(#)&quot; " ln="1" sev="5" auth="devtest" rule="PFO-04" tool="c++test" cat="PFO" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Implementation file 'MemoryLeak.cpp' should have the file name extension &quot;.cc&quot;" ln="1" sev="3" auth="devtest" rule="NAMING-38" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdio.h>' the corresponding C++ library header should be used" ln="1" sev="2" auth="devtest" rule="MISRA2008-18\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdio.h>' the corresponding C++ library header should be used" ln="1" sev="3" auth="devtest" rule="HICPP-1\_3\_3-a" tool="c++test" cat="HICPP-1\_3\_3" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdio.h>' the corresponding C++ library header should be used" ln="1" sev="2" auth="devtest" rule="AUTOSAR-A18\_0\_1-a" tool="c++test" cat="AUTOSAR-A18\_0\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdio.h>' the corresponding C++ library header should be used" ln="1" sev="3" auth="devtest" rule="CODSTA-CPP-59" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdio.h>' the corresponding C++ library header should be used" ln="1" sev="2" auth="devtest" rule="AUTOSAR-A1\_1\_1-d" tool="c++test" cat="AUTOSAR-A1\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The assertion density is lower than two assertions per function" ln="1" sev="3" auth="devtest" rule="METRICS-31" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The filename 'MemoryLeak.cpp' should be in lowercase" ln="1" sev="3" auth="devtest" rule="NAMING-03" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="All letters beside the first one in name 'IOException' should be lowercase" ln="3" sev="3" auth="devtest" rule="NAMING-40" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Class 'IOException' does not define any constructors" ln="3" sev="2" auth="devtest" rule="CODSTA-CPP-19" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Class 'IOException' missing assignment operator or special comment" ln="3" sev="3" auth="devtest" rule="MRM-04" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Class 'IOException' missing copy constructor or special comment" ln="3" sev="3" auth="devtest" rule="MRM-05" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Class 'IOException' should be final" ln="3" sev="4" auth="devtest" rule="AUTOSAR-A12\_4\_2-a" tool="c++test" cat="AUTOSAR-A12\_4\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Class 'IOException' should be final" ln="3" sev="3" auth="devtest" rule="CODSTA-MCPP-23" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Explicitly declare copy assignment operator in 'IOException'" ln="3" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Explicitly declare copy constructor in 'IOException'" ln="3" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Explicitly declare default constructor in 'IOException'" ln="3" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Explicitly declare destructor in 'IOException'" ln="3" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Explicitly declare move assignment operator in 'IOException'" ln="3" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Explicitly declare move constructor in 'IOException'" ln="3" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'IOException' identifier should have the 'C' prefix" ln="3" sev="3" auth="devtest" rule="NAMING-HN-19" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The declaration of the 'IOException' type should be preceded by a comment that contains the '@brief' tag" ln="3" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The declaration of the 'IOException' type should be preceded by a comment that contains the '@brief' tag" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Type 'IOException' is declared in global namespace" ln="3" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Type 'IOException' is declared in global namespace" ln="3" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Type 'IOException' is declared in global namespace" ln="3" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Type 'IOException' is declared in global namespace" ln="3" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="3" locStartPos="6" locEndLn="3" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare parameter 'file' as const" ln="5" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare parameter 'file' as const" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare parameter 'file' as const" ln="5" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare parameter 'file' as const" ln="5" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare parameter 'file' as const" ln="5" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has Cyclomatic Complexity value: 3" ln="5" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has Essential Complexity value: 1" ln="5" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has external linkage and is not declared in the header" ln="5" sev="4" auth="devtest" rule="OWASP2019-API9-e" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has external linkage and is not declared in the header" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A3\_3\_1-a" tool="c++test" cat="AUTOSAR-A3\_3\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has external linkage and is not declared in the header" ln="5" sev="4" auth="devtest" rule="JSF-137" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has external linkage and is not declared in the header" ln="5" sev="4" auth="devtest" rule="MISRA-023" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has external linkage and is not declared in the header" ln="5" sev="2" auth="devtest" rule="MISRA2008-3\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has external linkage and is not declared in the header" ln="5" sev="3" auth="devtest" rule="CERT\_C-DCL15-a" tool="c++test" cat="CERT\_C-DCL15" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' has external linkage and is not declared in the header" ln="5" sev="4" auth="devtest" rule="MISRA2004-8\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' returns a pointer type" ln="5" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Function 'readIntegerArray' returns a pointer type" ln="5" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Global function 'readIntegerArray' is declared in global namespace" ln="5" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Global function 'readIntegerArray' is declared in global namespace" ln="5" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Global function 'readIntegerArray' is declared in global namespace" ln="5" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Global function 'readIntegerArray' is declared in global namespace" ln="5" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Identifier name: 'file' differs only by case from its type name: 'FILE'" ln="5" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Identifier name: 'file' differs only by case from its type name: 'FILE'" ln="5" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Identifier name: 'file' differs only by case from its type name: 'FILE'" ln="5" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Identifier name: 'file' differs only by case from its type name: 'FILE'" ln="5" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Naming convention not followed: readIntegerArray" ln="5" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Parameter 'file' is not validated before use" ln="5" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Parameter 'file' is not validated before use" ln="5" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Parameter 'pSize' is not validated before use" ln="5" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Parameter 'pSize' is not validated before use" ln="5" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Pass parameter &quot;pSize&quot; with const specifier" ln="5" sev="2" auth="devtest" rule="AUTOSAR-M7\_1\_2-b" tool="c++test" cat="AUTOSAR-M7\_1\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Pass parameter &quot;pSize&quot; with const specifier" ln="5" sev="4" auth="devtest" rule="MISRAC2012-RULE\_8\_13-a" tool="c++test" cat="MISRAC2012-RULE\_8\_13" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Pass parameter &quot;pSize&quot; with const specifier" ln="5" sev="3" auth="devtest" rule="CERT\_C-DCL13-a" tool="c++test" cat="CERT\_C-DCL13" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Pass parameter &quot;pSize&quot; with const specifier" ln="5" sev="3" auth="devtest" rule="MISRA2004-16\_7" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Pass parameter &quot;pSize&quot; with const specifier" ln="5" sev="4" auth="devtest" rule="MISRA2012-RULE-8\_13\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Pass parameter &quot;pSize&quot; with const specifier" ln="5" sev="4" auth="devtest" rule="JSF-118" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Pass parameter &quot;pSize&quot; with const specifier" ln="5" sev="2" auth="devtest" rule="MISRA2008-7\_1\_2\_a" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Return type is not placed in line before function 'readIntegerArray'" ln="5" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'file' identifier should have the 'p' prefix" ln="5" sev="3" auth="devtest" rule="NAMING-HN-34" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'file' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="5" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'file' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'pSize' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-23" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'pSize' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-44" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'pSize' identifier should have the 'n' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-31" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'pSize' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="5" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'pSize' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'readIntegerArray' function is not used in the testing scope" ln="5" sev="3" auth="devtest" rule="GLOBAL-UNUSEDFUNC" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'readIntegerArray' function is not used in the testing scope" ln="5" sev="4" auth="devtest" rule="AUTOSAR-M0\_1\_10-a" tool="c++test" cat="AUTOSAR-M0\_1\_10" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'readIntegerArray' function is not used in the testing scope" ln="5" sev="2" auth="devtest" rule="MISRA2008-0\_1\_10\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'readIntegerArray' function should be preceded by a comment that contains the '@brief' tag" ln="5" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'readIntegerArray' function should be preceded by a comment that contains the '@brief' tag" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'readIntegerArray' function should be preceded by a comment that contains the '@return' tag" ln="5" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'readIntegerArray' function should be preceded by a comment that contains the '@return' tag" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The definition of the 'readIntegerArray' function is not preceded by a comment" ln="5" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The definition of the 'readIntegerArray' function is not preceded by a comment" ln="5" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The identifier 'file' differs only by case from identifier 'FILE' declared in file 'stdio.h'" ln="5" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The identifier 'file' differs only by case from identifier 'FILE' declared in file 'stdio.h'" ln="5" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The name 'readIntegerArray' should be composed only of lowercase letters" ln="5" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The name 'readIntegerArray' should be composed only of lowercase letters" ln="5" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared: file" ln="5" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared: pSize" ln="5" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The parameter of pointer type is declared: file" ln="5" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="28" locEndLn="5" locEndPos="29" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The parameter of pointer type is declared: pSize" ln="5" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="39" locEndLn="5" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The return type of the 'readIntegerArray' function should be declared as 'auto'" ln="5" sev="2" auth="devtest" rule="CODSTA-MCPP-08\_b" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="5" locEndLn="5" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="5" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="0" locEndLn="5" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="5" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="5" locStartPos="34" locEndLn="5" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="6" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="6" locStartPos="0" locEndLn="6" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<DupViol msg="Duplicated string: '&quot;%d&quot;'" ln="7" NvType="1" sev="3" auth="devtest" rule="CDD-DUPS" tool="c++test" cat="CDD" lang="cpp" locType="sr" config="1" hash="1013754779" NvActs="3" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="14" srcRngEndLn="7" srcRngEndPos="18" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="" desc="[Line 7] Duplicated string in file 'MemoryLeak.cpp'" sourceRngHash="1051585">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="19" srcRngEndLn="10" srcRngEndPos="23" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="" desc="[Line 10] Duplicated string in file 'MemoryLeak.cpp'" sourceRngHash="1051585">

<Props/>

</ElDesc>

</ElDescList>

</DupViol>

<StdViol msg="Non-ascii tab found" ln="7" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="7" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="7" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Prefer iostream.h to stdio.h" ln="7" sev="5" auth="devtest" rule="CODSTA-CPP-01" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: %d" ln="7" sev="5" auth="devtest" rule="CWE-798-a" tool="c++test" cat="CWE-798" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: %d" ln="7" sev="5" auth="devtest" rule="CODSTA-203" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: %d" ln="7" sev="1" auth="devtest" rule="APSC\_DV-003110-a" tool="c++test" cat="APSC\_DV-003110" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: %d" ln="7" sev="1" auth="devtest" rule="CERT\_C-MSC41-a" tool="c++test" cat="CERT\_C-MSC41" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal should not be passed as an argument of the 'const char \*' type in the 'fscanf' function call" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A27\_0\_4-d" tool="c++test" cat="AUTOSAR-A27\_0\_4" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal should not be passed as an argument of the 'const char \*' type in the 'fscanf' function call" ln="7" sev="3" auth="devtest" rule="PB-76" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="There should be number describing size of variable in '%d' on position:1" ln="7" sev="2" auth="devtest" rule="APSC\_DV-002390-c" tool="c++test" cat="APSC\_DV-002390" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="There should be number describing size of variable in '%d' on position:1" ln="7" sev="2" auth="devtest" rule="SECURITY-14" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unsafe string function 'fscanf' is being used" ln="7" sev="1" auth="devtest" rule="CERT\_C-STR07-a" tool="c++test" cat="CERT\_C-STR07" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unsafe string function 'fscanf' is being used" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A27\_0\_4-b" tool="c++test" cat="AUTOSAR-A27\_0\_4" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unsafe string function 'fscanf' is being used" ln="7" sev="3" auth="devtest" rule="CERT\_C-INT05-a" tool="c++test" cat="CERT\_C-INT05" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unsafe string function 'fscanf' is being used" ln="7" sev="2" auth="devtest" rule="SECURITY-13" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="3" auth="devtest" rule="CODSTA-122\_a" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="1" auth="devtest" rule="CERT\_C-ERR33-a" tool="c++test" cat="CERT\_C-ERR33" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="1" auth="devtest" rule="CERT\_C-POS54-a" tool="c++test" cat="CERT\_C-POS54" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="2" auth="devtest" rule="MISRAC2012-RULE\_17\_7-a" tool="c++test" cat="MISRAC2012-RULE\_17\_7" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="3" auth="devtest" rule="CERT\_C-EXP12-a" tool="c++test" cat="CERT\_C-EXP12" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="2" auth="devtest" rule="MISRA2012-RULE-17\_7\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="3" auth="devtest" rule="MISRA2004-16\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="2" auth="devtest" rule="AUTOSAR-M0\_3\_2-a" tool="c++test" cat="AUTOSAR-M0\_3\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="2" auth="devtest" rule="MISRA2008-0\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's &quot;fscanf&quot; return value" ln="7" sev="3" auth="devtest" rule="JSF-115" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's 'fscanf' return value" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A0\_1\_2-a" tool="c++test" cat="AUTOSAR-A0\_1\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's 'fscanf' return value" ln="7" sev="3" auth="devtest" rule="CODSTA-CPP-58" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's 'fscanf' return value" ln="7" sev="2" auth="devtest" rule="MISRA2008-0\_1\_7" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unused function's 'fscanf' return value" ln="7" sev="4" auth="devtest" rule="JSF-115\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="7" sev="2" auth="devtest" rule="MISRAC2012-RULE\_21\_6-a" tool="c++test" cat="MISRAC2012-RULE\_21\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="7" sev="2" auth="devtest" rule="MISRA2012-RULE-21\_6" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="7" sev="3" auth="devtest" rule="CERT\_C-ERR02-a" tool="c++test" cat="CERT\_C-ERR02" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="7" sev="3" auth="devtest" rule="CODSTA-110" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="7" sev="2" auth="devtest" rule="CERT\_C-ERR07-b" tool="c++test" cat="CERT\_C-ERR07" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="7" locStartPos="1" locEndLn="7" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'new' operator should not be used" ln="8" sev="2" auth="devtest" rule="MISRA2012-DIR-4\_12" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'new' operator should not be used" ln="8" sev="2" auth="devtest" rule="MISRA2008-18\_4\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'new' operator should not be used" ln="8" sev="2" auth="devtest" rule="MISRAC2012-RULE\_21\_3-a" tool="c++test" cat="MISRAC2012-RULE\_21\_3" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'new' operator should not be used" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A18\_5\_2-a" tool="c++test" cat="AUTOSAR-A18\_5\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'new' operator should not be used" ln="8" sev="2" auth="devtest" rule="JSF-206" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'new' operator should not be used" ln="8" sev="2" auth="devtest" rule="MISRA2012-RULE-21\_3" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'new' operator should not be used" ln="8" sev="3" auth="devtest" rule="MISRA2004-20\_4" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'new' operator should not be used" ln="8" sev="2" auth="devtest" rule="MISRAC2012-DIR\_4\_12-a" tool="c++test" cat="MISRAC2012-DIR\_4\_12" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Check the return value of new" ln="8" sev="3" auth="devtest" rule="MRM-34" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Check the return value of new" ln="8" sev="1" auth="devtest" rule="CERT\_CPP-MEM52-a" tool="c++test" cat="CERT\_CPP-MEM52" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<FlowViol msg="Data obtained from a file (&quot;\*pSize&quot;) is used to determine the size of memory allocation" ln="8" ruleSAFMsg="Point where the array is allocated" auth="devtest" sev="2" rule="BD-SECURITY-TDALLOC" ruleSCSCMsg="Tainting point" tool="c++test" id="-862528737" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="0" locEndLn="9" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Tainted data"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType=".C" desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

<Anns>

<Ann msg="Tainting point" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".P" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props/>

<Anns>

<Ann msg="Point where the array is allocated" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Data obtained from a file (&quot;\*pSize&quot;) is used to determine the size of memory allocation" ln="8" ruleSAFMsg="Point where the array is allocated" auth="devtest" sev="2" rule="CWE-770-a" ruleSCSCMsg="Tainting point" tool="c++test" id="-1466367756" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="0" locEndLn="9" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Tainted data"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType=".C" desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

<Anns>

<Ann msg="Tainting point" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".P" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props/>

<Anns>

<Ann msg="Point where the array is allocated" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Data obtained from a file (&quot;\*pSize&quot;) is used to determine the size of memory allocation" ln="8" ruleSAFMsg="Point where the array is allocated" auth="devtest" sev="2" rule="OWASP2019-API4-a" ruleSCSCMsg="Tainting point" tool="c++test" id="1519224323" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="0" locEndLn="9" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Tainted data"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType=".C" desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

<Anns>

<Ann msg="Tainting point" kind="cause"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".P" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props/>

<Anns>

<Ann msg="Point where the array is allocated" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Declare local variable 'data' as const" ln="8" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare local variable 'data' as const" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare local variable 'data' as const" ln="8" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare local variable 'data' as const" ln="8" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Declare local variable 'data' as const" ln="8" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Naming convention not followed: data" ln="8" sev="3" auth="devtest" rule="NAMING-22" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="8" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="8" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="8" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'data' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="8" sev="3" auth="devtest" rule="NAMING-HN-23" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'data' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="8" sev="3" auth="devtest" rule="NAMING-HN-44" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'data' identifier should have the 'n' prefix followed by a capital letter or an underscore" ln="8" sev="3" auth="devtest" rule="NAMING-HN-31" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'data' identifier should have the 'p' prefix" ln="8" sev="3" auth="devtest" rule="NAMING-HN-34" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'data' identifier should have the 'prg' prefix" ln="8" sev="3" auth="devtest" rule="NAMING-HN-16" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'data' variable should be commented" ln="8" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'data' variable should be commented" ln="8" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'data' variable should be declared with the 'auto' type specifier" ln="8" sev="2" auth="devtest" rule="CODSTA-MCPP-08\_a" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="8" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The definition of the 'data' variable should contain a braced initializer" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A8\_5\_2-a" tool="c++test" cat="AUTOSAR-A8\_5\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The definition of the 'data' variable should contain a braced initializer" ln="8" sev="3" auth="devtest" rule="CODSTA-MCPP-38" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: data" ln="8" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The variable of pointer type is declared: data" ln="8" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="6" locEndLn="8" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="8" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="8" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="17" locEndLn="8" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use vector or string instead of dynamically allocated array" ln="8" sev="3" auth="devtest" rule="STL-10" tool="c++test" cat="STL" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="8" locStartPos="13" locEndLn="8" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'i' shall be declared as unsigned int or signed int" ln="9" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="A loop does not have a fixed upper nor lower bound" ln="9" sev="3" auth="devtest" rule="CODSTA-83" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="1" locEndLn="9" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<FlowViol msg="Data obtained from a file (&quot;\*pSize&quot;) is used in a loop condition" ln="9" ruleSAFMsg="Tainted data use in a loop condition" auth="devtest" sev="1" rule="BD-SECURITY-TDLOOP" ruleSCSCMsg="Tainting point" tool="c++test" id="2054996746" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="10" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Tainted data"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType=".C" desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props>

<Prop key="Tracked variables" val="pSize[]"/>

</Props>

<Anns>

<Ann msg="Tainting point" kind="cause"/>

<Ann msg="Tainted data: pSize[]" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType="." desc="int\* data = new int[\*pSize];" rngLn="8">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType=".P" desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props>

<Prop key="Tracked variables" val="pSize[]"/>

</Props>

<Anns>

<Ann msg="Tainted data: pSize[]" kind="var"/>

<Ann msg="Tainted data use in a loop condition" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Non-ascii tab found" ln="9" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="9" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="9" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="9" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="9" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Postfix operator applied to variable 'i'; prefer prefix type" ln="9" sev="3" auth="devtest" rule="OPT-04" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="29" locEndLn="9" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="9" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="9" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="9" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="9" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'i' variable should be commented" ln="9" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'i' variable should be commented" ln="9" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="9" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="9" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="9" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="9" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="9" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="9" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="9" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="9" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The definition of the 'i' variable should contain a braced initializer" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A8\_5\_2-a" tool="c++test" cat="AUTOSAR-A8\_5\_2" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The definition of the 'i' variable should contain a braced initializer" ln="9" sev="3" auth="devtest" rule="CODSTA-MCPP-38" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="9" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="6" locEndLn="9" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="User input was directly used as a loop boundary: pSize" ln="9" sev="1" auth="devtest" rule="APSC\_DV-002560-k" tool="c++test" cat="APSC\_DV-002560" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="17" locEndLn="9" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="User input was directly used as a loop boundary: pSize" ln="9" sev="1" auth="devtest" rule="APSC\_DV-002550-k" tool="c++test" cat="APSC\_DV-002550" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="9" locStartPos="17" locEndLn="9" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="User input was directly used as a loop boundary: pSize" ln="9" sev="2" auth="devtest" rule="APSC\_DV-001290-b" tool="c++test" cat="APSC\_DV-001290" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="17" locEndLn="9" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="User input was directly used as a loop boundary: pSize" ln="9" sev="2" auth="devtest" rule="SECURITY-38" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="17" locEndLn="9" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="User input was directly used as a loop boundary: pSize" ln="9" sev="2" auth="devtest" rule="CWE-20-j" tool="c++test" cat="CWE-20" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="17" locEndLn="9" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="User input was directly used as a loop boundary: pSize" ln="9" sev="2" auth="devtest" rule="APSC\_DV-002530-k" tool="c++test" cat="APSC\_DV-002530" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="17" locEndLn="9" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="User input was directly used as a loop boundary: pSize" ln="9" sev="2" auth="devtest" rule="APSC\_DV-002520-k" tool="c++test" cat="APSC\_DV-002520" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="9" locStartPos="17" locEndLn="9" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Avoid indexing non-array objects" ln="10" sev="3" auth="devtest" rule="MISRA2004-17\_4" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="26" locEndLn="10" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Avoid indexing non-array objects" ln="10" sev="2" auth="devtest" rule="MISRA2008-5\_0\_15" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="26" locEndLn="10" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Avoid indexing non-array objects" ln="10" sev="2" auth="devtest" rule="AUTOSAR-M5\_0\_15-a" tool="c++test" cat="AUTOSAR-M5\_0\_15" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="26" locEndLn="10" locEndPos="27" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="1" locEndLn="10" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="1" locEndLn="10" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="1" locEndLn="10" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="10" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="10" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Prefer iostream.h to stdio.h" ln="10" sev="5" auth="devtest" rule="CODSTA-CPP-01" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="10" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="10" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="10" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="10" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The 'if' statement doesn't have an 'else' clause" ln="10" sev="3" auth="devtest" rule="CODSTA-23" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="2" locEndLn="10" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: %d" ln="10" sev="5" auth="devtest" rule="CWE-798-a" tool="c++test" cat="CWE-798" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="19" locEndLn="10" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: %d" ln="10" sev="5" auth="devtest" rule="CODSTA-203" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="19" locEndLn="10" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: %d" ln="10" sev="1" auth="devtest" rule="APSC\_DV-003110-a" tool="c++test" cat="APSC\_DV-003110" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="10" locStartPos="19" locEndLn="10" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal is embedded directly in the code: %d" ln="10" sev="1" auth="devtest" rule="CERT\_C-MSC41-a" tool="c++test" cat="CERT\_C-MSC41" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="10" locStartPos="19" locEndLn="10" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal should not be passed as an argument of the 'const char \*' type in the 'fscanf' function call" ln="10" sev="2" auth="devtest" rule="AUTOSAR-A27\_0\_4-d" tool="c++test" cat="AUTOSAR-A27\_0\_4" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="19" locEndLn="10" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="The string literal should not be passed as an argument of the 'const char \*' type in the 'fscanf' function call" ln="10" sev="3" auth="devtest" rule="PB-76" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="19" locEndLn="10" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="There should be number describing size of variable in '%d' on position:1" ln="10" sev="2" auth="devtest" rule="APSC\_DV-002390-c" tool="c++test" cat="APSC\_DV-002390" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="There should be number describing size of variable in '%d' on position:1" ln="10" sev="2" auth="devtest" rule="SECURITY-14" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unsafe string function 'fscanf' is being used" ln="10" sev="1" auth="devtest" rule="CERT\_C-STR07-a" tool="c++test" cat="CERT\_C-STR07" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unsafe string function 'fscanf' is being used" ln="10" sev="2" auth="devtest" rule="AUTOSAR-A27\_0\_4-b" tool="c++test" cat="AUTOSAR-A27\_0\_4" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unsafe string function 'fscanf' is being used" ln="10" sev="3" auth="devtest" rule="CERT\_C-INT05-a" tool="c++test" cat="CERT\_C-INT05" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Unsafe string function 'fscanf' is being used" ln="10" sev="2" auth="devtest" rule="SECURITY-13" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="10" sev="2" auth="devtest" rule="MISRAC2012-RULE\_21\_6-a" tool="c++test" cat="MISRAC2012-RULE\_21\_6" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="10" sev="2" auth="devtest" rule="MISRA2012-RULE-21\_6" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="10" sev="3" auth="devtest" rule="CERT\_C-ERR02-a" tool="c++test" cat="CERT\_C-ERR02" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="10" sev="3" auth="devtest" rule="CODSTA-110" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Usage of 'fscanf' function is not allowed" ln="10" sev="2" auth="devtest" rule="CERT\_C-ERR07-b" tool="c++test" cat="CERT\_C-ERR07" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="10" locStartPos="6" locEndLn="10" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="1" locEndLn="11" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="1" locEndLn="11" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="1" locEndLn="11" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="2" locEndLn="11" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="2" locEndLn="11" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="2" locEndLn="11" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="c++ exception handling structure is used in function 'readIntegerArray'" ln="11" sev="2" auth="devtest" rule="JSF-208" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="3" locEndLn="11" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="c++ exception handling structure is used in function 'readIntegerArray'" ln="11" sev="2" auth="devtest" rule="EXCEPT-05" tool="c++test" cat="EXCEPT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="11" locStartPos="3" locEndLn="11" locEndPos="4" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="12" locStartPos="1" locEndLn="12" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="12" locStartPos="1" locEndLn="12" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="12" locStartPos="1" locEndLn="12" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="13" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="13" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="13" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="14" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="14" locStartPos="1" locEndLn="14" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="14" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="14" locStartPos="0" locEndLn="14" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="14" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="14" locStartPos="0" locEndLn="14" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<StdViol msg="Non-ascii tab found" ln="14" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="14" locStartPos="0" locEndLn="14" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp"/>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="2" rule="APSC\_DV-002000-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="-1842545282" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="2" rule="AUTOSAR-A15\_0\_2-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="1331699030" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="2" rule="AUTOSAR-A15\_1\_4-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="1332641371" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="1" rule="BD-RES-LEAKS" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="-1151239445" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="3" rule="CERT\_C-CON30-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="103891581" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="3" rule="CERT\_C-FIO22-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="-1112344312" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="3" rule="CERT\_C-FIO42-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="-1112174134" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="1" rule="CERT\_C-MEM00-e" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="116382779" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="3" rule="CERT\_C-MEM12-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="116285462" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="2" rule="CERT\_C-MEM31-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="114391825" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="2" rule="CERT\_C-WIN30-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="1055995355" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="3" rule="CERT\_CPP-ERR57-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="-1542295099" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="3" rule="CERT\_CPP-FIO51-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="1119006820" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="1" rule="CWE-772-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="2027493177" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="4" rule="MISRA2012-DIR-4\_13\_a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="-1431710396" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="2" rule="MISRA2012-RULE-22\_1" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="1251621320" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="4" rule="MISRAC2012-DIR\_4\_13-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="2127420791" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="2" rule="MISRAC2012-RULE\_22\_1-a" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="1043097931" lang="cpp" locType="sr" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="Memory not deallocated: data" ln="15" ruleSAFMsg="Point where allocated memory is lost" auth="devtest" sev="1" rule="OWASP2019-API4-b" ruleSCSCMsg="Point where memory is allocated" tool="c++test" id="-1966237325" lang="cpp" locType="sr" urgent="true" config="1" hash="1013754779" locStartln="15" locStartPos="0" locEndLn="16" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" FirstElSrcRngStartln="8" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="9" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp">

<Props>

<Prop key="Tracked variables" val="Allocated memory"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="7" ElType="." desc="fscanf(file, &quot;%d&quot;, pSize);" rngLn="7">

<Props/>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="8" ElType=".C" desc="int\* data = new int[\*pSize];" rngLn="8">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Point where memory is allocated" kind="cause"/>

<Ann msg="Allocated memory: data" kind="var"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="9" srcRngStartPos="0" srcRngEndLn="10" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="9" ElType="." desc="for (int i = 0; i &lt; \*pSize; i++) {" rngLn="9">

<Props/>

<Anns>

<Ann msg="Loop condition evaluation: (i &lt; \*pSize) (assuming true)" kind="condEval"/>

<Ann msg="Entering the loop" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="10" srcRngStartPos="0" srcRngEndLn="11" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="10" ElType="!" desc="if (fscanf(file, &quot;%d&quot;, &amp;data[i]) == EOF) {" rngLn="10">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (fscanf(...) == -1) (assuming true)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="11" ElType="!E" desc="throw IOException();" rngLn="11" thrownTypes="throwStatement" throwingMethod="">

<Props/>

<Anns>

<Ann msg="Throws an exception" kind="except"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="15" srcRngStartPos="0" srcRngEndLn="16" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" srcRnghash="1013754779" ln="15" ElType=".P" desc="}" rngLn="15">

<Props>

<Prop key="Tracked variables" val="data"/>

</Props>

<Anns>

<Ann msg="Allocated memory: data" kind="var"/>

<Ann msg="Point where allocated memory is lost" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Add comment containing the copyright information at the begin of file 'NullPointer.cpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-02" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'NullPointer.cpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'NullPointer.cpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-03" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'NullPointer.cpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Disallowed #include notation is being used: &quot;Point.hpp&quot;" ln="1" sev="2" auth="devtest" rule="PREPROC-09" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Disallowed #include notation is being used: &quot;Point.hpp&quot;" ln="1" sev="2" auth="devtest" rule="JSF-033" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Implementation file 'NullPointer.cpp' should declare a local constant string that begins from characters &quot;@(#)&quot; " ln="1" sev="5" auth="devtest" rule="PFO-04" tool="c++test" cat="PFO" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Implementation file 'NullPointer.cpp' should have the file name extension &quot;.cc&quot;" ln="1" sev="3" auth="devtest" rule="NAMING-38" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Not a proper header file (\*.h ) is being included: &quot;Point.hpp&quot;" ln="1" sev="3" auth="devtest" rule="JSF-032" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Not a proper header file (\*.h ) is being included: &quot;Point.hpp&quot;" ln="1" sev="3" auth="devtest" rule="PREPROC-08" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The assertion density is lower than two assertions per function" ln="1" sev="3" auth="devtest" rule="METRICS-31" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The filename 'NullPointer.cpp' should be in lowercase" ln="1" sev="3" auth="devtest" rule="NAMING-03" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Ensure that C Standard Library is securely used" ln="3" sev="4" auth="devtest" rule="HICPP-17\_2\_1-a" tool="c++test" cat="HICPP-17\_2\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdlib.h>' the corresponding C++ library header should be used" ln="3" sev="2" auth="devtest" rule="MISRA2008-18\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdlib.h>' the corresponding C++ library header should be used" ln="3" sev="3" auth="devtest" rule="HICPP-1\_3\_3-a" tool="c++test" cat="HICPP-1\_3\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdlib.h>' the corresponding C++ library header should be used" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A18\_0\_1-a" tool="c++test" cat="AUTOSAR-A18\_0\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdlib.h>' the corresponding C++ library header should be used" ln="3" sev="3" auth="devtest" rule="CODSTA-CPP-59" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Instead of C library header '&lt;stdlib.h>' the corresponding C++ library header should be used" ln="3" sev="2" auth="devtest" rule="AUTOSAR-A1\_1\_1-d" tool="c++test" cat="AUTOSAR-A1\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="3" locStartPos="0" locEndLn="3" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'argc' shall be declared as unsigned int or signed int" ln="5" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'main' shall be declared as unsigned int or signed int" ln="5" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Declaration of variable 'argv' contains more than one level of pointer indirection" ln="5" sev="3" auth="devtest" rule="CODSTA-89" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="25" locEndLn="5" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Declaration of variable 'argv' contains more than one level of pointer indirection" ln="5" sev="3" auth="devtest" rule="HICPP-8\_1\_1-a" tool="c++test" cat="HICPP-8\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="25" locEndLn="5" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Declare parameter 'argc' as const" ln="5" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Declare parameter 'argc' as const" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Declare parameter 'argc' as const" ln="5" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Declare parameter 'argc' as const" ln="5" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Declare parameter 'argc' as const" ln="5" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Function 'main' has Cyclomatic Complexity value: 2" ln="5" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Function 'main' has Essential Complexity value: 1" ln="5" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Parameter 'argv' is not validated before use" ln="5" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="25" locEndLn="5" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Parameter 'argv' is not validated before use" ln="5" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="25" locEndLn="5" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Parameter 'argv' with array type was found in 'main' function declaration" ln="5" sev="2" auth="devtest" rule="JSF-097\_c" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="25" locEndLn="5" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Parameter 'argv' with array type was found in 'main' function declaration" ln="5" sev="2" auth="devtest" rule="CODSTA-142" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="25" locEndLn="5" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Return type is not placed in line before function 'main'" ln="5" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'argc' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-22" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'argc' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-43" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'argc' identifier should have the 'n' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-30" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="13" locEndLn="5" locEndPos="14" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'argc' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="5" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'argc' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'argv' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="5" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'argv' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function does not contain a catch(...) to catch all unhandled exceptions" ln="5" sev="3" auth="devtest" rule="CERT\_CPP-ERR50-d" tool="c++test" cat="CERT\_CPP-ERR50" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function does not contain a catch(...) to catch all unhandled exceptions" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A15\_3\_3-a" tool="c++test" cat="AUTOSAR-A15\_3\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function does not contain a catch(...) to catch all unhandled exceptions" ln="5" sev="3" auth="devtest" rule="HICPP-15\_3\_2-b" tool="c++test" cat="HICPP-15\_3\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function does not contain a catch(...) to catch all unhandled exceptions" ln="5" sev="4" auth="devtest" rule="MISRA2008-15\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function does not contain a catch(...) to catch all unhandled exceptions" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A15\_5\_3-d" tool="c++test" cat="AUTOSAR-A15\_5\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function does not contain a catch(...) to catch all unhandled exceptions" ln="5" sev="3" auth="devtest" rule="EXCEPT-06" tool="c++test" cat="EXCEPT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function does not contain a catch(...) to catch all unhandled exceptions" ln="5" sev="2" auth="devtest" rule="MISRA2008-15\_5\_3\_d" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function should be preceded by a comment that contains the '@brief' tag" ln="5" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function should be preceded by a comment that contains the '@brief' tag" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function should be preceded by a comment that contains the '@return' tag" ln="5" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'main' function should be preceded by a comment that contains the '@return' tag" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The basic numerical type 'char' should not be used" ln="5" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="19" locEndLn="5" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The basic numerical type 'char' should not be used" ln="5" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="19" locEndLn="5" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="9" locEndLn="5" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="9" locEndLn="5" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The definition of the 'main' function is not preceded by a comment" ln="5" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The definition of the 'main' function is not preceded by a comment" ln="5" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The parameter of pointer or array type is declared: argv" ln="5" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="25" locEndLn="5" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The parameter of pointer type is declared: argv" ln="5" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="5" locStartPos="25" locEndLn="5" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="6" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="6" locStartPos="0" locEndLn="6" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="7" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="7" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="7" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="7" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Locally allocated memory is not deleted locally for : point" ln="7" sev="3" auth="devtest" rule="MRM-18" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Locally allocated memory is not deleted locally for : point" ln="7" sev="1" auth="devtest" rule="CERT\_C-MEM00-a" tool="c++test" cat="CERT\_C-MEM00" lang="cpp" locType="sr" urgent="true" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="7" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="7" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="7" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A4\_10\_1-b" tool="c++test" cat="AUTOSAR-A4\_10\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="7" sev="4" auth="devtest" rule="HICPP-2\_5\_3-a" tool="c++test" cat="HICPP-2\_5\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Prefer 'nullptr' to '0' as the null pointer value" ln="7" sev="4" auth="devtest" rule="CODSTA-MCPP-04" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'point' variable should be commented" ln="7" sev="3" auth="devtest" rule="JSF-132\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'point' variable should be commented" ln="7" sev="3" auth="devtest" rule="COMMENT-05" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The definition of the 'point' variable should contain a braced initializer" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A8\_5\_2-a" tool="c++test" cat="AUTOSAR-A8\_5\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The definition of the 'point' variable should contain a braced initializer" ln="7" sev="3" auth="devtest" rule="CODSTA-MCPP-38" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The identifier 'point' differs only by case from identifier 'Point' declared in file 'Point.hpp'" ln="7" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The identifier 'point' differs only by case from identifier 'Point' declared in file 'Point.hpp'" ln="7" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The variable of pointer or array type is declared: point" ln="7" sev="3" auth="devtest" rule="CODSTA-94" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The variable of pointer type is declared: point" ln="7" sev="3" auth="devtest" rule="CODSTA-95" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="8" locEndLn="7" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Variable 'point' of pointer type is initialized with '0'" ln="7" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_a" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Variable 'point' of pointer type is initialized with '0'" ln="7" sev="2" auth="devtest" rule="AUTOSAR-M4\_10\_2-a" tool="c++test" cat="AUTOSAR-M4\_10\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Variable 'point' of pointer type is initialized with '0'" ln="7" sev="2" auth="devtest" rule="MISRA2008-4\_10\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Variable 'point' of pointer type is initialized with '0'" ln="7" sev="3" auth="devtest" rule="CODSTA-CPP-63" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Variable 'point' of pointer type is initialized with '0'" ln="7" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-a" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Variable 'point' of pointer type is initialized with '0'" ln="7" sev="3" auth="devtest" rule="CODSTA-131" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Variable 'point' of pointer type is initialized with '0'" ln="7" sev="2" auth="devtest" rule="MISRA2012-RULE-11\_9\_b" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Variable 'point' of pointer type is initialized with '0'" ln="7" sev="2" auth="devtest" rule="MISRAC2012-RULE\_11\_9-b" tool="c++test" cat="MISRAC2012-RULE\_11\_9" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="7" locStartPos="16" locEndLn="7" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Literal constant '3' is used" ln="8" sev="3" auth="devtest" rule="JSF-151" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="12" locEndLn="8" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Literal constant '3' is used" ln="8" sev="3" auth="devtest" rule="CODSTA-26" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="12" locEndLn="8" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Literal constant '3' is used" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A5\_1\_1-a" tool="c++test" cat="AUTOSAR-A5\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="12" locEndLn="8" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Literal constant '3' is used" ln="8" sev="3" auth="devtest" rule="HICPP-5\_1\_1-a" tool="c++test" cat="HICPP-5\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="12" locEndLn="8" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="No value of command line should be trusted: argc" ln="8" sev="2" auth="devtest" rule="SECURITY-35" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="5" locEndLn="8" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="8" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="8" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="8" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="8" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="8" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="8" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="8" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="8" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="8" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="0" locEndLn="8" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="The 'if' statement doesn't have an 'else' clause" ln="8" sev="3" auth="devtest" rule="CODSTA-23" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="8" locStartPos="1" locEndLn="8" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'new' operator should not be used" ln="9" sev="2" auth="devtest" rule="MISRA2012-DIR-4\_12" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'new' operator should not be used" ln="9" sev="2" auth="devtest" rule="MISRA2008-18\_4\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'new' operator should not be used" ln="9" sev="2" auth="devtest" rule="MISRAC2012-RULE\_21\_3-a" tool="c++test" cat="MISRAC2012-RULE\_21\_3" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'new' operator should not be used" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A18\_5\_2-a" tool="c++test" cat="AUTOSAR-A18\_5\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'new' operator should not be used" ln="9" sev="2" auth="devtest" rule="JSF-206" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'new' operator should not be used" ln="9" sev="2" auth="devtest" rule="MISRA2012-RULE-21\_3" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'new' operator should not be used" ln="9" sev="3" auth="devtest" rule="MISRA2004-20\_4" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'new' operator should not be used" ln="9" sev="2" auth="devtest" rule="MISRAC2012-DIR\_4\_12-a" tool="c++test" cat="MISRAC2012-DIR\_4\_12" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Check the return value of new" ln="9" sev="3" auth="devtest" rule="MRM-34" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="2" locEndLn="9" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Check the return value of new" ln="9" sev="1" auth="devtest" rule="CERT\_CPP-MEM52-a" tool="c++test" cat="CERT\_CPP-MEM52" lang="cpp" locType="sr" urgent="true" config="1" hash="-1257393797" locStartln="9" locStartPos="2" locEndLn="9" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Literal constant '2' is used" ln="9" sev="3" auth="devtest" rule="JSF-151" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="45" locEndLn="9" locEndPos="46" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Literal constant '2' is used" ln="9" sev="3" auth="devtest" rule="CODSTA-26" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="45" locEndLn="9" locEndPos="46" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Literal constant '2' is used" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A5\_1\_1-a" tool="c++test" cat="AUTOSAR-A5\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="45" locEndLn="9" locEndPos="46" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Literal constant '2' is used" ln="9" sev="3" auth="devtest" rule="HICPP-5\_1\_1-a" tool="c++test" cat="HICPP-5\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="45" locEndLn="9" locEndPos="46" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="No value of command line should be trusted: argv" ln="9" sev="2" auth="devtest" rule="SECURITY-35" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="2" locEndLn="9" locEndPos="3" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="No value of command line should be trusted: argv" ln="9" sev="2" auth="devtest" rule="SECURITY-35" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="10" locEndLn="9" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="No value of command line should be trusted: argv" ln="9" sev="2" auth="devtest" rule="SECURITY-35" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="No value of command line should be trusted: argv" ln="9" sev="2" auth="devtest" rule="SECURITY-35" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="25" locEndLn="9" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="No value of command line should be trusted: argv" ln="9" sev="2" auth="devtest" rule="SECURITY-35" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="No value of command line should be trusted: argv" ln="9" sev="2" auth="devtest" rule="SECURITY-35" tool="c++test" cat="SECURITY" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="40" locEndLn="9" locEndPos="41" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="9" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="9" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="9" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="9" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="1" locEndLn="9" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="9" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="1" locEndLn="9" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="9" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="1" locEndLn="9" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="MISRAC2012-RULE\_21\_7-a" tool="c++test" cat="MISRAC2012-RULE\_21\_7" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="3" auth="devtest" rule="CERT\_CPP-ERR62-a" tool="c++test" cat="CERT\_CPP-ERR62" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="MISRA2008-18\_0\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="3" auth="devtest" rule="CERT\_C-ERR34-a" tool="c++test" cat="CERT\_C-ERR34" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="3" auth="devtest" rule="MISRA2004-20\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="3" auth="devtest" rule="CERT\_C-MSC24-a" tool="c++test" cat="CERT\_C-MSC24" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="CERT\_C-ERR07-a" tool="c++test" cat="CERT\_C-ERR07" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A18\_0\_2-b" tool="c++test" cat="AUTOSAR-A18\_0\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A18\_0\_2-a" tool="c++test" cat="AUTOSAR-A18\_0\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="MISRA2012-RULE-21\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="JSF-023" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="20" locEndLn="9" locEndPos="21" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="MISRAC2012-RULE\_21\_7-a" tool="c++test" cat="MISRAC2012-RULE\_21\_7" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="3" auth="devtest" rule="CERT\_CPP-ERR62-a" tool="c++test" cat="CERT\_CPP-ERR62" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="MISRA2008-18\_0\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="3" auth="devtest" rule="CERT\_C-ERR34-a" tool="c++test" cat="CERT\_C-ERR34" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="3" auth="devtest" rule="MISRA2004-20\_10" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="3" auth="devtest" rule="CERT\_C-MSC24-a" tool="c++test" cat="CERT\_C-MSC24" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="CERT\_C-ERR07-a" tool="c++test" cat="CERT\_C-ERR07" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A18\_0\_2-b" tool="c++test" cat="AUTOSAR-A18\_0\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A18\_0\_2-a" tool="c++test" cat="AUTOSAR-A18\_0\_2" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="MISRA2012-RULE-21\_7" tool="c++test" cat="MISRA2012-RULE" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Usage of 'atoi' function is not allowed" ln="9" sev="2" auth="devtest" rule="JSF-023" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="9" locStartPos="35" locEndLn="9" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="10" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="2" rule="AUTOSAR-A5\_3\_2-a" ruleSCSCMsg="Source of null value" tool="c++test" id="1390236584" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="2" rule="AUTOSAR-M0\_3\_1-f" ruleSCSCMsg="Source of null value" tool="c++test" id="78528533" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="1" rule="BD-PB-NP" ruleSCSCMsg="Source of null value" tool="c++test" id="476450621" lang="cpp" locType="sr" urgent="true" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="1" rule="CERT\_C-ERR33-c" ruleSCSCMsg="Source of null value" tool="c++test" id="2017494553" lang="cpp" locType="sr" urgent="true" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="1" rule="CERT\_C-EXP34-a" ruleSCSCMsg="Source of null value" tool="c++test" id="1982312066" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="3" rule="CERT\_C-MSC19-b" ruleSCSCMsg="Source of null value" tool="c++test" id="-799733766" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="1" rule="CERT\_C-POS54-c" ruleSCSCMsg="Source of null value" tool="c++test" id="-1180988405" lang="cpp" locType="sr" urgent="true" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="1" rule="CERT\_CPP-STR51-a" ruleSCSCMsg="Source of null value" tool="c++test" id="1665615379" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="1" rule="CWE-476-a" ruleSCSCMsg="Source of null value" tool="c++test" id="-1747021404" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="1" rule="HICPP-5\_2\_1-c" ruleSCSCMsg="Source of null value" tool="c++test" id="-15090486" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="5" rule="MISRA2008-0\_3\_1\_b" ruleSCSCMsg="Source of null value" tool="c++test" id="564686231" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="2" rule="MISRA2012-DIR-4\_1\_b" ruleSCSCMsg="Source of null value" tool="c++test" id="-2094766674" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<FlowViol msg="&quot;point&quot; may possibly be null" ln="11" ruleSAFMsg="Null pointer dereferencing point" auth="devtest" sev="2" rule="MISRAC2012-DIR\_4\_1-b" ruleSCSCMsg="Source of null value" tool="c++test" id="-1597100009" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="12" locEndPos="0" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" FirstElSrcRngStartln="7" FirstElSrcRngStartPos="0" FirstElSrcRngEndLn="8" FirstElSrcRngEndPos="0" FirstElSrcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp">

<Props>

<Prop key="Tracked variables" val="Null value carrier"/>

</Props>

<ElDescList>

<ElDesc srcRngStartln="7" srcRngStartPos="0" srcRngEndLn="8" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="7" ElType=".C" desc="Point\* point = 0;" rngLn="7">

<Props/>

<Anns>

<Ann msg="Source of null value" kind="cause"/>

<Ann msg="Null value carrier: point" kind="comment"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="8" srcRngStartPos="0" srcRngEndLn="9" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="8" ElType="." desc="if (argc > 3) {" rngLn="8">

<Props/>

<Anns>

<Ann msg="Condition evaluation: (argc > 3) (assuming false)" kind="condEval"/>

</Anns>

</ElDesc>

<ElDesc srcRngStartln="11" srcRngStartPos="0" srcRngEndLn="12" srcRngEndPos="0" srcRngFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" srcRnghash="-1257393797" ln="11" ElType=".P" desc="point->reflectAcrossX();" rngLn="11">

<Props/>

<Anns>

<Ann msg="Null value carrier: point" kind="comment"/>

<Ann msg="Null pointer dereferencing point" kind="point"/>

</Anns>

</ElDesc>

</ElDescList>

</FlowViol>

<StdViol msg="Non-ascii tab found" ln="11" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="11" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="11" locStartPos="0" locEndLn="11" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="12" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="13" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="13" locStartPos="1" locEndLn="13" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="13" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="13" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="Non-ascii tab found" ln="13" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="-1257393797" locStartln="13" locStartPos="0" locEndLn="13" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp"/>

<StdViol msg="'Point.hpp' filename extension does not adhere to naming conventions" ln="1" sev="3" auth="devtest" rule="NAMING-43" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'Point.hpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-02" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'Point.hpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'Point.hpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-03" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'Point.hpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="File 'Point.hpp' should have the &quot;.icc&quot; file name extension" ln="1" sev="3" auth="devtest" rule="NAMING-39" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Header file 'Point.hpp' should have the file name extension &quot;.h&quot;" ln="1" sev="3" auth="devtest" rule="JSF-053" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Header file 'Point.hpp' should have the file name extension &quot;.h&quot;" ln="1" sev="3" auth="devtest" rule="NAMING-41" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Header file 'Point.hpp' should have the file name extension &quot;.hh&quot;" ln="1" sev="3" auth="devtest" rule="NAMING-37" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="POINT\_HPP macro name should appear as first #ifndef or #if preprocessor directive" ln="1" sev="3" auth="devtest" rule="PFO-07" tool="c++test" cat="PFO" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The assertion density is lower than two assertions per function" ln="1" sev="3" auth="devtest" rule="METRICS-31" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The filename 'Point.hpp' should be in lowercase" ln="1" sev="3" auth="devtest" rule="NAMING-03" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Class 'Point' missing assignment operator or special comment" ln="4" sev="3" auth="devtest" rule="MRM-04" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Class 'Point' missing assignment operator or special comment" ln="4" sev="3" auth="devtest" rule="MRM-47" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Class 'Point' missing copy constructor or special comment" ln="4" sev="3" auth="devtest" rule="MRM-05" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Class 'Point' missing copy constructor or special comment" ln="4" sev="3" auth="devtest" rule="MRM-48" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Class 'Point' should be final" ln="4" sev="4" auth="devtest" rule="AUTOSAR-A12\_4\_2-a" tool="c++test" cat="AUTOSAR-A12\_4\_2" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Class 'Point' should be final" ln="4" sev="3" auth="devtest" rule="CODSTA-MCPP-23" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Explicitly declare copy assignment operator in 'Point'" ln="4" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Explicitly declare copy constructor in 'Point'" ln="4" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Explicitly declare default constructor in 'Point'" ln="4" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Explicitly declare destructor in 'Point'" ln="4" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Explicitly declare move assignment operator in 'Point'" ln="4" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Explicitly declare move constructor in 'Point'" ln="4" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Opening '{' and closing '}' should be in the same column" ln="4" sev="3" auth="devtest" rule="FORMAT-34" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="0" locEndLn="4" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Public section must be before protected and private sections" ln="4" sev="3" auth="devtest" rule="JSF-057\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Public section must be before protected and private sections" ln="4" sev="3" auth="devtest" rule="CODSTA-CPP-46" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="4" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="0" locEndLn="4" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'Point' identifier should have the 'C' prefix" ln="4" sev="3" auth="devtest" rule="NAMING-HN-19" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The declaration of the 'Point' type should be preceded by a comment that contains the '@brief' tag" ln="4" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The declaration of the 'Point' type should be preceded by a comment that contains the '@brief' tag" ln="4" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'Point' differs only by case from identifier 'point' declared in file 'NullPointer.cpp'" ln="4" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'Point' differs only by case from identifier 'point' declared in file 'NullPointer.cpp'" ln="4" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'Point' differs only by case from identifier 'point' declared in file 'Point.hpp'" ln="4" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'Point' differs only by case from identifier 'point' declared in file 'Point.hpp'" ln="4" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'Point' differs only by case from identifier 'point' declared in file 'Shapes.hpp'" ln="4" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'Point' differs only by case from identifier 'point' declared in file 'Shapes.hpp'" ln="4" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Type 'Point' is declared in global namespace" ln="4" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Type 'Point' is declared in global namespace" ln="4" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Type 'Point' is declared in global namespace" ln="4" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Type 'Point' is declared in global namespace" ln="4" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="4" locStartPos="6" locEndLn="4" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="'\_x' shall be declared as unsigned int or signed int" ln="5" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="'\_y' shall be declared as unsigned int or signed int" ln="5" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate declaration statement" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-c" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate declaration statement" ln="5" sev="3" auth="devtest" rule="FORMAT-33" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate declaration statement" ln="5" sev="2" auth="devtest" rule="MISRA2008-8\_0\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate declaration statement" ln="5" sev="3" auth="devtest" rule="HICPP-7\_1\_1-b" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate declaration statement" ln="5" sev="2" auth="devtest" rule="AUTOSAR-M8\_0\_1-a" tool="c++test" cat="AUTOSAR-M8\_0\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate declaration statement" ln="5" sev="3" auth="devtest" rule="CERT\_C-DCL04-a" tool="c++test" cat="CERT\_C-DCL04" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate line" ln="5" sev="2" auth="devtest" rule="JSF-152" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate line" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_7-b" tool="c++test" cat="AUTOSAR-A7\_1\_7" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate line" ln="5" sev="3" auth="devtest" rule="HICPP-7\_1\_1-a" tool="c++test" cat="HICPP-7\_1\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare variable '\_y' in a separate line" ln="5" sev="3" auth="devtest" rule="FORMAT-29" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Member variable '\_x' shall begin with a lowercase letter" ln="5" sev="3" auth="devtest" rule="NAMING-07" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Member variable '\_y' shall begin with a lowercase letter" ln="5" sev="3" auth="devtest" rule="NAMING-07" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Naming convention not followed: \_x" ln="5" sev="3" auth="devtest" rule="NAMING-14" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Naming convention not followed: \_x" ln="5" sev="3" auth="devtest" rule="NAMING-19" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Naming convention not followed: \_y" ln="5" sev="3" auth="devtest" rule="NAMING-14" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Naming convention not followed: \_y" ln="5" sev="3" auth="devtest" rule="NAMING-19" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_x' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-22" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_x' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-43" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_x' identifier should have the 'm\_' prefix" ln="5" sev="3" auth="devtest" rule="NAMING-HN-29" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_x' identifier should have the 'n' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-30" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_x' member variable should be preceded by a comment that contains the '@brief' tag" ln="5" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_x' member variable should be preceded by a comment that contains the '@brief' tag" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_y' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-22" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_y' identifier should have the 'i' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-43" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_y' identifier should have the 'm\_' prefix" ln="5" sev="3" auth="devtest" rule="NAMING-HN-29" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_y' identifier should have the 'n' prefix followed by a capital letter or an underscore" ln="5" sev="3" auth="devtest" rule="NAMING-HN-30" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_y' member variable should be preceded by a comment that contains the '@brief' tag" ln="5" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The '\_y' member variable should be preceded by a comment that contains the '@brief' tag" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="5" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier '\_x' differs only by presence/absence of the underscore character from identifier 'x' declared in file 'Point.hpp'" ln="5" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier '\_x' differs only by presence/absence of the underscore character from identifier 'x' declared in file 'Point.hpp'" ln="5" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier '\_y' differs only by presence/absence of the underscore character from identifier 'y' declared in file 'Point.hpp'" ln="5" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier '\_y' differs only by presence/absence of the underscore character from identifier 'y' declared in file 'Point.hpp'" ln="5" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="5" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="5" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="4" locEndLn="5" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_x' is not allowed" ln="5" sev="3" auth="devtest" rule="NAMING-33" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_x' is not allowed" ln="5" sev="3" auth="devtest" rule="JSF-047" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="8" locEndLn="5" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_y' is not allowed" ln="5" sev="3" auth="devtest" rule="NAMING-33" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_y' is not allowed" ln="5" sev="3" auth="devtest" rule="JSF-047" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="5" locStartPos="12" locEndLn="5" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="'x' shall be declared as unsigned int or signed int" ln="7" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="'y' shall be declared as unsigned int or signed int" ln="7" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Class 'Point' defines an inline constructor" ln="7" sev="3" auth="devtest" rule="OPT-17" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'x' as const" ln="7" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'x' as const" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'x' as const" ln="7" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'x' as const" ln="7" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'x' as const" ln="7" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'y' as const" ln="7" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'y' as const" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'y' as const" ln="7" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'y' as const" ln="7" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Declare parameter 'y' as const" ln="7" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Function 'Point' has Cyclomatic Complexity value: 1" ln="7" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Function 'Point' has Essential Complexity value: 1" ln="7" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'x' differs only by presence/absence of the underscore character from identifier name: '\_x' declared in class" ln="7" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'x' differs only by presence/absence of the underscore character from identifier name: '\_x' declared in class" ln="7" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'x' differs only by presence/absence of the underscore character from identifier name: '\_x' declared in class" ln="7" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'x' differs only by presence/absence of the underscore character from identifier name: '\_x' declared in class" ln="7" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'y' differs only by presence/absence of the underscore character from identifier name: '\_y' declared in class" ln="7" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'y' differs only by presence/absence of the underscore character from identifier name: '\_y' declared in class" ln="7" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'y' differs only by presence/absence of the underscore character from identifier name: '\_y' declared in class" ln="7" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'y' differs only by presence/absence of the underscore character from identifier name: '\_y' declared in class" ln="7" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="7" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="7" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="7" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="7" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="40" locEndLn="7" locEndPos="41" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="7" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="40" locEndLn="7" locEndPos="41" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="7" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="40" locEndLn="7" locEndPos="41" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="7" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="7" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="7" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'Point' function should be declared 'noexcept'" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A15\_4\_4-a" tool="c++test" cat="AUTOSAR-A15\_4\_4" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'Point' function should be declared 'noexcept'" ln="7" sev="3" auth="devtest" rule="CODSTA-MCPP-09" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'Point' function should be preceded by a comment that contains the '@brief' tag" ln="7" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'Point' function should be preceded by a comment that contains the '@brief' tag" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'x' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="7" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'x' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'y' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="7" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'y' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="7" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The definition of the 'Point' function is not preceded by a comment" ln="7" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The definition of the 'Point' function is not preceded by a comment" ln="7" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'Point' differs only by case from identifier 'point' declared in file 'Point.hpp'" ln="7" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'Point' differs only by case from identifier 'point' declared in file 'Point.hpp'" ln="7" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="4" locEndLn="7" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'x' differs only by presence/absence of the underscore character from identifier '\_x' declared in file 'Point.hpp'" ln="7" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'x' differs only by presence/absence of the underscore character from identifier '\_x' declared in file 'Point.hpp'" ln="7" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="14" locEndLn="7" locEndPos="15" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'y' differs only by presence/absence of the underscore character from identifier '\_y' declared in file 'Point.hpp'" ln="7" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'y' differs only by presence/absence of the underscore character from identifier '\_y' declared in file 'Point.hpp'" ln="7" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="21" locEndLn="7" locEndPos="22" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="7" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="10" locEndLn="7" locEndPos="11" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="7" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="7" locStartPos="17" locEndLn="7" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Function 'reflectAcrossX' has Cyclomatic Complexity value: 1" ln="8" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Function 'reflectAcrossX' has Essential Complexity value: 1" ln="8" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Naming convention not followed: reflectAcrossX" ln="8" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Return type is not placed in line before function 'reflectAcrossX'" ln="8" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'reflectAcrossX' function should be declared 'noexcept'" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A15\_4\_4-a" tool="c++test" cat="AUTOSAR-A15\_4\_4" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'reflectAcrossX' function should be declared 'noexcept'" ln="8" sev="3" auth="devtest" rule="CODSTA-MCPP-09" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'reflectAcrossX' function should be preceded by a comment that contains the '@brief' tag" ln="8" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'reflectAcrossX' function should be preceded by a comment that contains the '@brief' tag" ln="8" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The definition of the 'reflectAcrossX' function is not preceded by a comment" ln="8" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The definition of the 'reflectAcrossX' function is not preceded by a comment" ln="8" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The name 'reflectAcrossX' should be composed only of lowercase letters" ln="8" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The name 'reflectAcrossX' should be composed only of lowercase letters" ln="8" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="8" locStartPos="9" locEndLn="8" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="9" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="9" locStartPos="0" locEndLn="9" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Function 'translate' has Cyclomatic Complexity value: 1" ln="13" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Function 'translate' has Essential Complexity value: 1" ln="13" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Naming convention not followed: translate" ln="13" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Parameter 'vector' is not validated before use" ln="13" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="32" locEndLn="13" locEndPos="33" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Parameter 'vector' is not validated before use" ln="13" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="32" locEndLn="13" locEndPos="33" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Return type is not placed in line before function 'translate'" ln="13" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Small object of type 'Point' with trivial copy constructor should be passed by value" ln="13" sev="3" auth="devtest" rule="HICPP-8\_2\_3-b" tool="c++test" cat="HICPP-8\_2\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Small object of type 'Point' with trivial copy constructor should be passed by value" ln="13" sev="2" auth="devtest" rule="AUTOSAR-A8\_4\_7-b" tool="c++test" cat="AUTOSAR-A8\_4\_7" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'const' qualifier should be placed on the right hand side of the type" ln="13" sev="3" auth="devtest" rule="FORMAT-47\_a" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="19" locEndLn="13" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'const' qualifier should be placed on the right hand side of the type" ln="13" sev="3" auth="devtest" rule="HICPP-7\_1\_4-a" tool="c++test" cat="HICPP-7\_1\_4" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="19" locEndLn="13" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'translate' function should be declared 'noexcept'" ln="13" sev="2" auth="devtest" rule="AUTOSAR-A15\_4\_4-a" tool="c++test" cat="AUTOSAR-A15\_4\_4" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'translate' function should be declared 'noexcept'" ln="13" sev="3" auth="devtest" rule="CODSTA-MCPP-09" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'translate' function should be preceded by a comment that contains the '@brief' tag" ln="13" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'translate' function should be preceded by a comment that contains the '@brief' tag" ln="13" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'vector' identifier should have the 'k' prefix" ln="13" sev="3" auth="devtest" rule="NAMING-HN-11" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="32" locEndLn="13" locEndPos="33" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'vector' identifier should have the 'r' prefix" ln="13" sev="3" auth="devtest" rule="NAMING-HN-35" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="32" locEndLn="13" locEndPos="33" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'vector' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="13" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'vector' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="13" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The definition of the 'translate' function is not preceded by a comment" ln="13" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The definition of the 'translate' function is not preceded by a comment" ln="13" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="13" locStartPos="9" locEndLn="13" locEndPos="10" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="14" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="14" locStartPos="0" locEndLn="14" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="15" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="15" locStartPos="4" locEndLn="15" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="15" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="15" locStartPos="4" locEndLn="15" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="15" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="15" locStartPos="4" locEndLn="15" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="16" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="16" locStartPos="4" locEndLn="16" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="16" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="16" locStartPos="4" locEndLn="16" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="16" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="16" locStartPos="4" locEndLn="16" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="'squareDistanceTo' shall be declared as unsigned int or signed int" ln="19" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Function 'squareDistanceTo' has Cyclomatic Complexity value: 1" ln="19" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Function 'squareDistanceTo' has Essential Complexity value: 1" ln="19" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from identifier name: 'Point' ( line: '4')" ln="19" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from identifier name: 'Point' ( line: '4')" ln="19" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from identifier name: 'Point' ( line: '4')" ln="19" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from identifier name: 'Point' ( line: '4')" ln="19" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from identifier name: 'Point' declared in class" ln="19" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from identifier name: 'Point' declared in class" ln="19" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from identifier name: 'Point' declared in class" ln="19" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from identifier name: 'Point' declared in class" ln="19" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="19" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="19" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="19" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="19" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Member function 'squareDistanceTo' should be declared as const" ln="19" sev="3" auth="devtest" rule="CODSTA-CPP-54" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Member function 'squareDistanceTo' should be declared as const" ln="19" sev="3" auth="devtest" rule="JSF-069" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Member function 'squareDistanceTo' should be declared as const" ln="19" sev="3" auth="devtest" rule="HICPP-9\_1\_1-a" tool="c++test" cat="HICPP-9\_1\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Member function 'squareDistanceTo' should be declared as const" ln="19" sev="2" auth="devtest" rule="AUTOSAR-M9\_3\_3-a" tool="c++test" cat="AUTOSAR-M9\_3\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Member function 'squareDistanceTo' should be declared as const" ln="19" sev="3" auth="devtest" rule="CODSTA-CPP-78" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Member function 'squareDistanceTo' should be declared as const" ln="19" sev="2" auth="devtest" rule="MISRA2008-9\_3\_3" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Naming convention not followed: squareDistanceTo" ln="19" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Parameter 'point' is not validated before use" ln="19" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Parameter 'point' is not validated before use" ln="19" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Return type is not placed in line before function 'squareDistanceTo'" ln="19" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Small object of type 'Point' with trivial copy constructor should be passed by value" ln="19" sev="3" auth="devtest" rule="HICPP-8\_2\_3-b" tool="c++test" cat="HICPP-8\_2\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Small object of type 'Point' with trivial copy constructor should be passed by value" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A8\_4\_7-b" tool="c++test" cat="AUTOSAR-A8\_4\_7" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'const' qualifier should be placed on the right hand side of the type" ln="19" sev="3" auth="devtest" rule="FORMAT-47\_a" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="25" locEndLn="19" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'const' qualifier should be placed on the right hand side of the type" ln="19" sev="3" auth="devtest" rule="HICPP-7\_1\_4-a" tool="c++test" cat="HICPP-7\_1\_4" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="25" locEndLn="19" locEndPos="26" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'point' identifier should have the 'k' prefix" ln="19" sev="3" auth="devtest" rule="NAMING-HN-11" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'point' identifier should have the 'r' prefix" ln="19" sev="3" auth="devtest" rule="NAMING-HN-35" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'point' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="19" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'point' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'squareDistanceTo' function should be declared 'noexcept'" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A15\_4\_4-a" tool="c++test" cat="AUTOSAR-A15\_4\_4" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'squareDistanceTo' function should be declared 'noexcept'" ln="19" sev="3" auth="devtest" rule="CODSTA-MCPP-09" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'squareDistanceTo' function should be preceded by a comment that contains the '@brief' tag" ln="19" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'squareDistanceTo' function should be preceded by a comment that contains the '@brief' tag" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'squareDistanceTo' function should be preceded by a comment that contains the '@return' tag" ln="19" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The 'squareDistanceTo' function should be preceded by a comment that contains the '@return' tag" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="19" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="19" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="19" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="19" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="19" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="19" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="19" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="19" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The definition of the 'squareDistanceTo' function is not preceded by a comment" ln="19" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The definition of the 'squareDistanceTo' function is not preceded by a comment" ln="19" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'point' differs only by case from identifier 'Point' declared in file 'Point.hpp'" ln="19" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The identifier 'point' differs only by case from identifier 'Point' declared in file 'Point.hpp'" ln="19" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="38" locEndLn="19" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The name 'squareDistanceTo' should be composed only of lowercase letters" ln="19" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The name 'squareDistanceTo' should be composed only of lowercase letters" ln="19" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="The return type of the 'squareDistanceTo' function should be declared as 'auto'" ln="19" sev="2" auth="devtest" rule="CODSTA-MCPP-08\_b" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="8" locEndLn="19" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="19" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="20" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="20" locStartPos="0" locEndLn="20" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="21" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="21" locStartPos="5" locEndLn="21" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Line has over 79 characters" ln="21" sev="3" auth="devtest" rule="FORMAT-04" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="21" locStartPos="0" locEndLn="21" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="21" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="21" locStartPos="4" locEndLn="21" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="21" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="21" locStartPos="4" locEndLn="21" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Non-ascii tab found" ln="21" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="21" locStartPos="4" locEndLn="21" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use // comments only" ln="25" sev="3" auth="devtest" rule="HICPP-2\_3\_1-a" tool="c++test" cat="HICPP-2\_3\_1" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="25" locStartPos="7" locEndLn="25" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use // comments only" ln="25" sev="2" auth="devtest" rule="JSF-126" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="25" locStartPos="7" locEndLn="25" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="Use // comments only" ln="25" sev="3" auth="devtest" rule="COMMENT-01" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1950870755" locStartln="25" locStartPos="7" locEndLn="25" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp"/>

<StdViol msg="'Shapes.hpp' filename extension does not adhere to naming conventions" ln="1" sev="3" auth="devtest" rule="NAMING-43" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'Shapes.hpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-02" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Add comment containing the copyright information at the begin of file 'Shapes.hpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'Shapes.hpp'" ln="1" sev="3" auth="devtest" rule="COMMENT-03" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Add comment containing the information on the file at the begin of file 'Shapes.hpp'" ln="1" sev="3" auth="devtest" rule="JSF-133\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="File 'Shapes.hpp' should have the &quot;.icc&quot; file name extension" ln="1" sev="3" auth="devtest" rule="NAMING-39" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="File 'Shapes.hpp' contains more than one class definition" ln="1" sev="3" auth="devtest" rule="PFO-03" tool="c++test" cat="PFO" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Header file 'Shapes.hpp' contains more than one class declaration: 'Shape, LineSegment, Circle'" ln="1" sev="3" auth="devtest" rule="PFO-10" tool="c++test" cat="PFO" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Header file 'Shapes.hpp' should have the file name extension &quot;.h&quot;" ln="1" sev="3" auth="devtest" rule="JSF-053" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Header file 'Shapes.hpp' should have the file name extension &quot;.h&quot;" ln="1" sev="3" auth="devtest" rule="NAMING-41" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Header file 'Shapes.hpp' should have the file name extension &quot;.hh&quot;" ln="1" sev="3" auth="devtest" rule="NAMING-37" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="SHAPES\_HPP macro name should appear as first #ifndef or #if preprocessor directive" ln="1" sev="3" auth="devtest" rule="PFO-07" tool="c++test" cat="PFO" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The assertion density is lower than two assertions per function" ln="1" sev="3" auth="devtest" rule="METRICS-31" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The filename 'Shapes.hpp' should be in lowercase" ln="1" sev="3" auth="devtest" rule="NAMING-03" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="1" locStartPos="0" locEndLn="1" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Disallowed #include notation is being used: &quot;Point.hpp&quot;" ln="4" sev="2" auth="devtest" rule="PREPROC-09" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="4" locStartPos="0" locEndLn="4" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Disallowed #include notation is being used: &quot;Point.hpp&quot;" ln="4" sev="2" auth="devtest" rule="JSF-033" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="4" locStartPos="0" locEndLn="4" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Not a proper header file (\*.h ) is being included: &quot;Point.hpp&quot;" ln="4" sev="3" auth="devtest" rule="JSF-032" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="4" locStartPos="0" locEndLn="4" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Not a proper header file (\*.h ) is being included: &quot;Point.hpp&quot;" ln="4" sev="3" auth="devtest" rule="PREPROC-08" tool="c++test" cat="PREPROC" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="4" locStartPos="0" locEndLn="4" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' has virtual functions without a virtual destructor" ln="6" sev="2" auth="devtest" rule="OOP-23" tool="c++test" cat="OOP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' missing assignment operator or special comment" ln="6" sev="3" auth="devtest" rule="MRM-04" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' missing assignment operator or special comment" ln="6" sev="3" auth="devtest" rule="MRM-47" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' missing copy constructor or special comment" ln="6" sev="3" auth="devtest" rule="MRM-05" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' missing copy constructor or special comment" ln="6" sev="3" auth="devtest" rule="MRM-48" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' must define a virtual destructor" ln="6" sev="1" auth="devtest" rule="OOP-22" tool="c++test" cat="OOP" lang="cpp" locType="sr" urgent="true" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' must define a virtual destructor" ln="6" sev="1" auth="devtest" rule="CWE-772-b" tool="c++test" cat="CWE-772" lang="cpp" locType="sr" urgent="true" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' must define a virtual destructor" ln="6" sev="2" auth="devtest" rule="CERT\_CPP-OOP52-a" tool="c++test" cat="CERT\_CPP-OOP52" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' must define a virtual destructor" ln="6" sev="2" auth="devtest" rule="JSF-078" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' should be final" ln="6" sev="4" auth="devtest" rule="AUTOSAR-A12\_4\_2-a" tool="c++test" cat="AUTOSAR-A12\_4\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' should be final" ln="6" sev="3" auth="devtest" rule="CODSTA-MCPP-23" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare copy assignment operator in 'Shape'" ln="6" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare copy constructor in 'Shape'" ln="6" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare default constructor in 'Shape'" ln="6" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare destructor in 'Shape'" ln="6" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare move assignment operator in 'Shape'" ln="6" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare move constructor in 'Shape'" ln="6" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="In a class 'Shape' number of accessor functions (2) exceeds number of member variables (1)" ln="6" sev="4" auth="devtest" rule="OPT-27" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' should be in the same column" ln="6" sev="3" auth="devtest" rule="FORMAT-34" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="0" locEndLn="6" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Protected section must be before private section" ln="6" sev="3" auth="devtest" rule="JSF-057\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Protected section must be before private section" ln="6" sev="3" auth="devtest" rule="CODSTA-CPP-47" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public section must be before protected and private sections" ln="6" sev="3" auth="devtest" rule="JSF-057\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public section must be before protected and private sections" ln="6" sev="3" auth="devtest" rule="CODSTA-CPP-46" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put base class 'Shape' into separate file" ln="6" sev="3" auth="devtest" rule="CODSTA-CPP-12" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="6" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="0" locEndLn="6" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'Shape' identifier should have the 'C' prefix" ln="6" sev="3" auth="devtest" rule="NAMING-HN-19" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The declaration of the 'Shape' type should be preceded by a comment that contains the '@brief' tag" ln="6" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The declaration of the 'Shape' type should be preceded by a comment that contains the '@brief' tag" ln="6" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'Shape' is declared in global namespace" ln="6" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'Shape' is declared in global namespace" ln="6" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'Shape' is declared in global namespace" ln="6" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'Shape' is declared in global namespace" ln="6" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="6" locStartPos="6" locEndLn="6" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Member variable '\_position' shall begin with a lowercase letter" ln="7" sev="3" auth="devtest" rule="NAMING-07" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: \_position" ln="7" sev="3" auth="devtest" rule="NAMING-14" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="7" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="7" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="7" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="0" locEndLn="7" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_position' identifier should have the 'm\_' prefix" ln="7" sev="3" auth="devtest" rule="NAMING-HN-29" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_position' member variable should be preceded by a comment that contains the '@brief' tag" ln="7" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_position' member variable should be preceded by a comment that contains the '@brief' tag" ln="7" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier '\_position' differs only by presence/absence of the underscore character from identifier 'position' declared in file 'Shapes.hpp'" ln="7" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier '\_position' differs only by presence/absence of the underscore character from identifier 'position' declared in file 'Shapes.hpp'" ln="7" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_position' is not allowed" ln="7" sev="3" auth="devtest" rule="NAMING-33" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_position' is not allowed" ln="7" sev="3" auth="devtest" rule="JSF-047" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="7" locStartPos="7" locEndLn="7" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Make virtual functions nonpublic, and public functions nonvirtual: getArea" ln="9" sev="3" auth="devtest" rule="CODSTA-CPP-25" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: getArea" ln="9" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@brief' tag" ln="9" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@brief' tag" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@return' tag" ln="9" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@return' tag" ln="9" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="9" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="12" locEndLn="9" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="9" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="12" locEndLn="9" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="9" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="12" locEndLn="9" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="9" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="12" locEndLn="9" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="9" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="12" locEndLn="9" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="9" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="12" locEndLn="9" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="9" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="12" locEndLn="9" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="9" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="12" locEndLn="9" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The declaration of the 'getArea' function is not preceded by a comment" ln="9" sev="4" auth="devtest" rule="JSF-134\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The declaration of the 'getArea' function is not preceded by a comment" ln="9" sev="3" auth="devtest" rule="COMMENT-04\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The name 'getArea' should be composed only of lowercase letters" ln="9" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The name 'getArea' should be composed only of lowercase letters" ln="9" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="9" locStartPos="19" locEndLn="9" locEndPos="20" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="10" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="27" locEndLn="10" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getPosition' has Cyclomatic Complexity value: 1" ln="10" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getPosition' has Essential Complexity value: 1" ln="10" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Member function 'getPosition' returns handles to member data: '\_position'" ln="10" sev="3" auth="devtest" rule="CODSTA-CPP-06" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="27" locEndLn="10" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: getPosition" ln="10" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="10" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="10" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' should be in the same column" ln="10" sev="3" auth="devtest" rule="FORMAT-34" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="10" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public member function 'getPosition' returns non-const handles to member data: '\_position'" ln="10" sev="2" auth="devtest" rule="MISRA2008-9\_3\_2\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="27" locEndLn="10" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public member function 'getPosition' returns non-const handles to member data: '\_position'" ln="10" sev="3" auth="devtest" rule="HICPP-9\_1\_4-a" tool="c++test" cat="HICPP-9\_1\_4" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="27" locEndLn="10" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public member function 'getPosition' returns non-const handles to member data: '\_position'" ln="10" sev="3" auth="devtest" rule="OOP-36" tool="c++test" cat="OOP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="27" locEndLn="10" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public member function 'getPosition' returns non-const handles to member data: '\_position'" ln="10" sev="2" auth="devtest" rule="AUTOSAR-A9\_3\_1-a" tool="c++test" cat="AUTOSAR-A9\_3\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="27" locEndLn="10" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="10" sev="3" auth="devtest" rule="FORMAT-03" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="45" locEndLn="10" locEndPos="46" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="10" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="45" locEndLn="10" locEndPos="46" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="10" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="45" locEndLn="10" locEndPos="46" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="10" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="45" locEndLn="10" locEndPos="46" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="10" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="10" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="10" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="10" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="0" locEndLn="10" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Return type is not placed in line before function 'getPosition'" ln="10" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getPosition' function should be declared 'noexcept'" ln="10" sev="2" auth="devtest" rule="AUTOSAR-A15\_4\_4-a" tool="c++test" cat="AUTOSAR-A15\_4\_4" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getPosition' function should be declared 'noexcept'" ln="10" sev="3" auth="devtest" rule="CODSTA-MCPP-09" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getPosition' function should be preceded by a comment that contains the '@brief' tag" ln="10" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getPosition' function should be preceded by a comment that contains the '@brief' tag" ln="10" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getPosition' function should be preceded by a comment that contains the '@return' tag" ln="10" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getPosition' function should be preceded by a comment that contains the '@return' tag" ln="10" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'getPosition' function is not preceded by a comment" ln="10" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'getPosition' function is not preceded by a comment" ln="10" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The name 'getPosition' should be composed only of lowercase letters" ln="10" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The name 'getPosition' should be composed only of lowercase letters" ln="10" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The return type of the 'getPosition' function should be declared as 'auto&amp;'" ln="10" sev="2" auth="devtest" rule="CODSTA-MCPP-08\_b" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="10" locStartPos="11" locEndLn="10" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Shape' defines an inline constructor" ln="12" sev="3" auth="devtest" rule="OPT-17" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Constructor 'Shape' allowing conversion should be made explicit" ln="12" sev="4" auth="devtest" rule="JSF-177\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Constructor 'Shape' allowing conversion should be made explicit" ln="12" sev="1" auth="devtest" rule="CODSTA-CPP-04" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" urgent="true" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Constructor 'Shape' allowing conversion should be made explicit" ln="12" sev="1" auth="devtest" rule="HICPP-12\_1\_1-a" tool="c++test" cat="HICPP-12\_1\_1" lang="cpp" locType="sr" urgent="true" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="12" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="12" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="12" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="12" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="12" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'Shape' has Cyclomatic Complexity value: 1" ln="12" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'Shape' has Essential Complexity value: 1" ln="12" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in class" ln="12" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in class" ln="12" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in class" ln="12" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in class" ln="12" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="12" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="12" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="12" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="12" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="12" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="12" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="49" locEndLn="12" locEndPos="50" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="12" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="12" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="12" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="0" locEndLn="12" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'Shape' function should be preceded by a comment that contains the '@brief' tag" ln="12" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'Shape' function should be preceded by a comment that contains the '@brief' tag" ln="12" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'position' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="12" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'position' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="12" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'Shape' function is not preceded by a comment" ln="12" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'Shape' function is not preceded by a comment" ln="12" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="4" locEndLn="12" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'position' differs only by presence/absence of the underscore character from identifier '\_position' declared in file 'Shapes.hpp'" ln="12" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'position' differs only by presence/absence of the underscore character from identifier '\_position' declared in file 'Shapes.hpp'" ln="12" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'Shape' is passed by value" ln="12" sev="3" auth="devtest" rule="OPT-14" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'Shape' is passed by value" ln="12" sev="4" auth="devtest" rule="AUTOSAR-A8\_4\_3-a" tool="c++test" cat="AUTOSAR-A8\_4\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'Shape' is passed by value" ln="12" sev="4" auth="devtest" rule="JSF-125\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'Shape' is passed by value" ln="12" sev="4" auth="devtest" rule="JSF-117" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="12" locStartPos="16" locEndLn="12" locEndPos="17" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="A declaration of the 'LineSegment' type which is not used has been found" ln="15" sev="4" auth="devtest" rule="AUTOSAR-A0\_1\_6-a" tool="c++test" cat="AUTOSAR-A0\_1\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="A declaration of the 'LineSegment' type which is not used has been found" ln="15" sev="3" auth="devtest" rule="GLOBAL-UNUSEDTYPE" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="A declaration of the 'LineSegment' type which is not used has been found" ln="15" sev="2" auth="devtest" rule="MISRA2008-0\_1\_5" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="All letters beside the first one in name 'LineSegment' should be lowercase" ln="15" sev="3" auth="devtest" rule="NAMING-40" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'LineSegment' has virtual functions without a virtual destructor" ln="15" sev="2" auth="devtest" rule="OOP-23" tool="c++test" cat="OOP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'LineSegment' missing assignment operator or special comment" ln="15" sev="3" auth="devtest" rule="MRM-04" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'LineSegment' missing assignment operator or special comment" ln="15" sev="3" auth="devtest" rule="MRM-47" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'LineSegment' missing copy constructor or special comment" ln="15" sev="3" auth="devtest" rule="MRM-05" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'LineSegment' missing copy constructor or special comment" ln="15" sev="3" auth="devtest" rule="MRM-48" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'LineSegment' should be final" ln="15" sev="4" auth="devtest" rule="AUTOSAR-A12\_4\_2-a" tool="c++test" cat="AUTOSAR-A12\_4\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'LineSegment' should be final" ln="15" sev="3" auth="devtest" rule="CODSTA-MCPP-23" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare copy assignment operator in 'LineSegment'" ln="15" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare copy constructor in 'LineSegment'" ln="15" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare default constructor in 'LineSegment'" ln="15" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare destructor in 'LineSegment'" ln="15" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare move assignment operator in 'LineSegment'" ln="15" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare move constructor in 'LineSegment'" ln="15" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' should be in the same column" ln="15" sev="3" auth="devtest" rule="FORMAT-34" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public section must be before protected and private sections" ln="15" sev="3" auth="devtest" rule="JSF-057\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public section must be before protected and private sections" ln="15" sev="3" auth="devtest" rule="CODSTA-CPP-46" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="15" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="0" locEndLn="15" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'LineSegment' identifier should have the 'C' prefix" ln="15" sev="3" auth="devtest" rule="NAMING-HN-19" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The declaration of the 'LineSegment' type should be preceded by a comment that contains the '@brief' tag" ln="15" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The declaration of the 'LineSegment' type should be preceded by a comment that contains the '@brief' tag" ln="15" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'LineSegment' is declared in global namespace" ln="15" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'LineSegment' is declared in global namespace" ln="15" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'LineSegment' is declared in global namespace" ln="15" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'LineSegment' is declared in global namespace" ln="15" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="15" locStartPos="6" locEndLn="15" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Member variable '\_end' shall begin with a lowercase letter" ln="16" sev="3" auth="devtest" rule="NAMING-07" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: \_end" ln="16" sev="3" auth="devtest" rule="NAMING-14" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="16" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="0" locEndLn="16" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="16" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="0" locEndLn="16" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="16" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="0" locEndLn="16" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Private member variable '\_end' is declared but not used" ln="16" sev="3" auth="devtest" rule="OPT-05" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Private member variable '\_end' is declared but not used" ln="16" sev="2" auth="devtest" rule="AUTOSAR-M0\_1\_3-c" tool="c++test" cat="AUTOSAR-M0\_1\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Private member variable '\_end' is declared but not used" ln="16" sev="2" auth="devtest" rule="MISRA2008-0\_1\_3\_c" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_end' identifier should have the 'm\_' prefix" ln="16" sev="3" auth="devtest" rule="NAMING-HN-29" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_end' member variable should be preceded by a comment that contains the '@brief' tag" ln="16" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_end' member variable should be preceded by a comment that contains the '@brief' tag" ln="16" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier '\_end' differs only by presence/absence of the underscore character from identifier 'end' declared in file 'Shapes.hpp'" ln="16" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier '\_end' differs only by presence/absence of the underscore character from identifier 'end' declared in file 'Shapes.hpp'" ln="16" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_end' is not allowed" ln="16" sev="3" auth="devtest" rule="NAMING-33" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_end' is not allowed" ln="16" sev="3" auth="devtest" rule="JSF-047" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="16" locStartPos="7" locEndLn="16" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'LineSegment' defines an inline constructor" ln="18" sev="3" auth="devtest" rule="OPT-17" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'end' as const" ln="18" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'end' as const" ln="18" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'end' as const" ln="18" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'end' as const" ln="18" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'end' as const" ln="18" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="18" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="18" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="18" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="18" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="18" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'LineSegment' has Cyclomatic Complexity value: 1" ln="18" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'LineSegment' has Essential Complexity value: 1" ln="18" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'end' differs only by presence/absence of the underscore character from identifier name: '\_end' declared in class" ln="18" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'end' differs only by presence/absence of the underscore character from identifier name: '\_end' declared in class" ln="18" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'end' differs only by presence/absence of the underscore character from identifier name: '\_end' declared in class" ln="18" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'end' differs only by presence/absence of the underscore character from identifier name: '\_end' declared in class" ln="18" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in base class" ln="18" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in base class" ln="18" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in base class" ln="18" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in base class" ln="18" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="18" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="0" locEndLn="18" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="18" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="0" locEndLn="18" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="18" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="0" locEndLn="18" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="18" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="73" locEndLn="18" locEndPos="74" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="18" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="73" locEndLn="18" locEndPos="74" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="18" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="73" locEndLn="18" locEndPos="74" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="18" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="0" locEndLn="18" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="18" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="0" locEndLn="18" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="18" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="0" locEndLn="18" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'LineSegment' function is not used in the testing scope" ln="18" sev="3" auth="devtest" rule="GLOBAL-UNUSEDFUNC" tool="c++test" cat="GLOBAL" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'LineSegment' function is not used in the testing scope" ln="18" sev="4" auth="devtest" rule="AUTOSAR-M0\_1\_10-a" tool="c++test" cat="AUTOSAR-M0\_1\_10" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'LineSegment' function is not used in the testing scope" ln="18" sev="2" auth="devtest" rule="MISRA2008-0\_1\_10\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'LineSegment' function should be preceded by a comment that contains the '@brief' tag" ln="18" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'LineSegment' function should be preceded by a comment that contains the '@brief' tag" ln="18" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'end' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="18" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'end' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="18" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'position' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="18" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'position' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="18" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'LineSegment' function is not preceded by a comment" ln="18" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'LineSegment' function is not preceded by a comment" ln="18" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="4" locEndLn="18" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'end' differs only by presence/absence of the underscore character from identifier '\_end' declared in file 'Shapes.hpp'" ln="18" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'end' differs only by presence/absence of the underscore character from identifier '\_end' declared in file 'Shapes.hpp'" ln="18" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'position' differs only by presence/absence of the underscore character from identifier '\_position' declared in file 'Shapes.hpp'" ln="18" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'position' differs only by presence/absence of the underscore character from identifier '\_position' declared in file 'Shapes.hpp'" ln="18" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'end' of function 'LineSegment' is passed by value" ln="18" sev="3" auth="devtest" rule="OPT-14" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'end' of function 'LineSegment' is passed by value" ln="18" sev="4" auth="devtest" rule="AUTOSAR-A8\_4\_3-a" tool="c++test" cat="AUTOSAR-A8\_4\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'end' of function 'LineSegment' is passed by value" ln="18" sev="4" auth="devtest" rule="JSF-125\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'end' of function 'LineSegment' is passed by value" ln="18" sev="4" auth="devtest" rule="JSF-117" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="38" locEndLn="18" locEndPos="39" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'LineSegment' is passed by value" ln="18" sev="3" auth="devtest" rule="OPT-14" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'LineSegment' is passed by value" ln="18" sev="4" auth="devtest" rule="AUTOSAR-A8\_4\_3-a" tool="c++test" cat="AUTOSAR-A8\_4\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'LineSegment' is passed by value" ln="18" sev="4" auth="devtest" rule="JSF-125\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'LineSegment' is passed by value" ln="18" sev="4" auth="devtest" rule="JSF-117" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="18" locStartPos="22" locEndLn="18" locEndPos="23" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="19" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="29" locEndLn="19" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declaration of virtual function 'getArea' should have 'override' or 'final' specifier" ln="19" sev="3" auth="devtest" rule="CODSTA-MCPP-24" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declaration of virtual function 'getArea' should have 'override' or 'final' specifier" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A10\_3\_1-a" tool="c++test" cat="AUTOSAR-A10\_3\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getArea' has Cyclomatic Complexity value: 1" ln="19" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getArea' has Essential Complexity value: 1" ln="19" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getArea' has declared return type 'double' but returns value of type 'int'" ln="19" sev="3" auth="devtest" rule="PB-05" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="29" locEndLn="19" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getArea' is both virtual and inline" ln="19" sev="3" auth="devtest" rule="OOP-25" tool="c++test" cat="OOP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Implicit conversion from integral to floating type in function's return" ln="19" sev="2" auth="devtest" rule="MISRA2008-5\_0\_5\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="29" locEndLn="19" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Implicit conversion from integral to floating type in function's return" ln="19" sev="3" auth="devtest" rule="CERT\_C-EXP39-a" tool="c++test" cat="CERT\_C-EXP39" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="29" locEndLn="19" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Implicit conversion from integral to floating type in function's return" ln="19" sev="2" auth="devtest" rule="AUTOSAR-M5\_0\_5-a" tool="c++test" cat="AUTOSAR-M5\_0\_5" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="29" locEndLn="19" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Implicit conversion from integral to floating type in function's return" ln="19" sev="3" auth="devtest" rule="MISRA2004-10\_1\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="29" locEndLn="19" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Make virtual functions nonpublic, and public functions nonvirtual: getArea" ln="19" sev="3" auth="devtest" rule="CODSTA-CPP-25" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: getArea" ln="19" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="19" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="19" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' should be in the same column" ln="19" sev="3" auth="devtest" rule="FORMAT-34" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="19" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="19" sev="3" auth="devtest" rule="FORMAT-03" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="39" locEndLn="19" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="19" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="39" locEndLn="19" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="19" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="39" locEndLn="19" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="19" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="39" locEndLn="19" locEndPos="40" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="19" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="19" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="19" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="19" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="0" locEndLn="19" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Return type is not placed in line before function 'getArea'" ln="19" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be declared with the 'override' specifier" ln="19" sev="3" auth="devtest" rule="CODSTA-MCPP-05" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be declared with the 'override' specifier" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A10\_3\_2-a" tool="c++test" cat="AUTOSAR-A10\_3\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be declared with the 'override' specifier" ln="19" sev="3" auth="devtest" rule="HICPP-10\_2\_1-a" tool="c++test" cat="HICPP-10\_2\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@brief' tag" ln="19" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@brief' tag" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@return' tag" ln="19" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@return' tag" ln="19" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="19" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="19" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="19" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="19" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="19" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="19" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="19" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="19" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="4" locEndLn="19" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'getArea' function is not preceded by a comment" ln="19" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'getArea' function is not preceded by a comment" ln="19" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The name 'getArea' should be composed only of lowercase letters" ln="19" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The name 'getArea' should be composed only of lowercase letters" ln="19" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use the virtual keyword for 'getArea' function" ln="19" sev="3" auth="devtest" rule="OOP-21" tool="c++test" cat="OOP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use the virtual keyword for 'getArea' function" ln="19" sev="2" auth="devtest" rule="MISRA2008-10\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="19" locStartPos="11" locEndLn="19" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Circle' has virtual functions without a virtual destructor" ln="22" sev="2" auth="devtest" rule="OOP-23" tool="c++test" cat="OOP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Circle' missing assignment operator or special comment" ln="22" sev="3" auth="devtest" rule="MRM-04" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Circle' missing assignment operator or special comment" ln="22" sev="3" auth="devtest" rule="MRM-47" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Circle' missing copy constructor or special comment" ln="22" sev="3" auth="devtest" rule="MRM-05" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Circle' missing copy constructor or special comment" ln="22" sev="3" auth="devtest" rule="MRM-48" tool="c++test" cat="MRM" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Circle' should be final" ln="22" sev="4" auth="devtest" rule="AUTOSAR-A12\_4\_2-a" tool="c++test" cat="AUTOSAR-A12\_4\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Circle' should be final" ln="22" sev="3" auth="devtest" rule="CODSTA-MCPP-23" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare copy assignment operator in 'Circle'" ln="22" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare copy constructor in 'Circle'" ln="22" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare default constructor in 'Circle'" ln="22" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare destructor in 'Circle'" ln="22" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare move assignment operator in 'Circle'" ln="22" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Explicitly declare move constructor in 'Circle'" ln="22" sev="3" auth="devtest" rule="HICPP-12\_5\_1-a" tool="c++test" cat="HICPP-12\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' should be in the same column" ln="22" sev="3" auth="devtest" rule="FORMAT-34" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="0" locEndLn="22" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public section must be before protected and private sections" ln="22" sev="3" auth="devtest" rule="JSF-057\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Public section must be before protected and private sections" ln="22" sev="3" auth="devtest" rule="CODSTA-CPP-46" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="22" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="0" locEndLn="22" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The declaration of the 'Circle' type should be preceded by a comment that contains the '@brief' tag" ln="22" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The declaration of the 'Circle' type should be preceded by a comment that contains the '@brief' tag" ln="22" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'Circle' is declared in global namespace" ln="22" sev="4" auth="devtest" rule="JSF-098" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'Circle' is declared in global namespace" ln="22" sev="3" auth="devtest" rule="CODSTA-CPP-36" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'Circle' is declared in global namespace" ln="22" sev="2" auth="devtest" rule="AUTOSAR-M7\_3\_1-a" tool="c++test" cat="AUTOSAR-M7\_3\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Type 'Circle' is declared in global namespace" ln="22" sev="2" auth="devtest" rule="MISRA2008-7\_3\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="22" locStartPos="6" locEndLn="22" locEndPos="7" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Member variable '\_radius' shall begin with a lowercase letter" ln="23" sev="3" auth="devtest" rule="NAMING-07" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: \_radius" ln="23" sev="3" auth="devtest" rule="NAMING-14" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: \_radius" ln="23" sev="3" auth="devtest" rule="NAMING-15" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="23" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="0" locEndLn="23" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="23" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="0" locEndLn="23" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="23" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="0" locEndLn="23" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_radius' identifier should have the 'd' prefix followed by a capital letter or an underscore" ln="23" sev="3" auth="devtest" rule="NAMING-HN-12" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_radius' identifier should have the 'm\_' prefix" ln="23" sev="3" auth="devtest" rule="NAMING-HN-29" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_radius' member variable should be preceded by a comment that contains the '@brief' tag" ln="23" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The '\_radius' member variable should be preceded by a comment that contains the '@brief' tag" ln="23" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="23" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="1" locEndLn="23" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="23" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="1" locEndLn="23" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="23" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="1" locEndLn="23" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="23" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="1" locEndLn="23" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="23" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="1" locEndLn="23" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="23" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="1" locEndLn="23" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="23" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="1" locEndLn="23" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="23" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="1" locEndLn="23" locEndPos="2" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier '\_radius' differs only by presence/absence of the underscore character from identifier 'radius' declared in file 'Shapes.hpp'" ln="23" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier '\_radius' differs only by presence/absence of the underscore character from identifier 'radius' declared in file 'Shapes.hpp'" ln="23" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_radius' is not allowed" ln="23" sev="3" auth="devtest" rule="NAMING-33" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Using underscore at the beginning of the name '\_radius' is not allowed" ln="23" sev="3" auth="devtest" rule="JSF-047" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="23" locStartPos="8" locEndLn="23" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Class 'Circle' defines an inline constructor" ln="25" sev="3" auth="devtest" rule="OPT-17" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="25" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="25" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="25" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="25" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'position' as const" ln="25" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'radius' as const" ln="25" sev="3" auth="devtest" rule="CERT\_C-DCL00-a" tool="c++test" cat="CERT\_C-DCL00" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'radius' as const" ln="25" sev="2" auth="devtest" rule="AUTOSAR-A7\_1\_1-a" tool="c++test" cat="AUTOSAR-A7\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'radius' as const" ln="25" sev="2" auth="devtest" rule="MISRA2008-7\_1\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'radius' as const" ln="25" sev="3" auth="devtest" rule="CODSTA-CPP-53" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declare parameter 'radius' as const" ln="25" sev="3" auth="devtest" rule="HICPP-7\_1\_2-a" tool="c++test" cat="HICPP-7\_1\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'Circle' has Cyclomatic Complexity value: 1" ln="25" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'Circle' has Essential Complexity value: 1" ln="25" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in base class" ln="25" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in base class" ln="25" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in base class" ln="25" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'position' differs only by presence/absence of the underscore character from identifier name: '\_position' declared in base class" ln="25" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'radius' differs only by presence/absence of the underscore character from identifier name: '\_radius' declared in class" ln="25" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'radius' differs only by presence/absence of the underscore character from identifier name: '\_radius' declared in class" ln="25" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'radius' differs only by presence/absence of the underscore character from identifier name: '\_radius' declared in class" ln="25" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'radius' differs only by presence/absence of the underscore character from identifier name: '\_radius' declared in class" ln="25" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="25" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="0" locEndLn="25" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="25" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="0" locEndLn="25" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="25" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="0" locEndLn="25" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="25" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="78" locEndLn="25" locEndPos="79" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="25" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="78" locEndLn="25" locEndPos="79" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="25" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="78" locEndLn="25" locEndPos="79" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="25" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="0" locEndLn="25" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="25" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="0" locEndLn="25" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="25" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="0" locEndLn="25" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'Circle' function should be preceded by a comment that contains the '@brief' tag" ln="25" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'Circle' function should be preceded by a comment that contains the '@brief' tag" ln="25" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'position' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="25" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'position' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="25" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'radius' identifier should have the 'd' prefix followed by a capital letter or an underscore" ln="25" sev="3" auth="devtest" rule="NAMING-HN-12" tool="c++test" cat="NAMING-HN" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'radius' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="25" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'radius' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="25" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="25" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="27" locEndLn="25" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="25" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="27" locEndLn="25" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="25" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="27" locEndLn="25" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="25" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="27" locEndLn="25" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="25" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="27" locEndLn="25" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="25" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="27" locEndLn="25" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="25" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="27" locEndLn="25" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="25" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="27" locEndLn="25" locEndPos="28" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'Circle' function is not preceded by a comment" ln="25" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'Circle' function is not preceded by a comment" ln="25" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="4" locEndLn="25" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'position' differs only by presence/absence of the underscore character from identifier '\_position' declared in file 'Shapes.hpp'" ln="25" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'position' differs only by presence/absence of the underscore character from identifier '\_position' declared in file 'Shapes.hpp'" ln="25" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'radius' differs only by presence/absence of the underscore character from identifier '\_radius' declared in file 'Shapes.hpp'" ln="25" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'radius' differs only by presence/absence of the underscore character from identifier '\_radius' declared in file 'Shapes.hpp'" ln="25" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="34" locEndLn="25" locEndPos="35" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'Circle' is passed by value" ln="25" sev="3" auth="devtest" rule="OPT-14" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'Circle' is passed by value" ln="25" sev="4" auth="devtest" rule="AUTOSAR-A8\_4\_3-a" tool="c++test" cat="AUTOSAR-A8\_4\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'Circle' is passed by value" ln="25" sev="4" auth="devtest" rule="JSF-125\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'position' of function 'Circle' is passed by value" ln="25" sev="4" auth="devtest" rule="JSF-117" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="25" locStartPos="17" locEndLn="25" locEndPos="18" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="26" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="29" locEndLn="26" locEndPos="30" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declaration of virtual function 'getArea' should have 'override' or 'final' specifier" ln="26" sev="3" auth="devtest" rule="CODSTA-MCPP-24" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Declaration of virtual function 'getArea' should have 'override' or 'final' specifier" ln="26" sev="2" auth="devtest" rule="AUTOSAR-A10\_3\_1-a" tool="c++test" cat="AUTOSAR-A10\_3\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getArea' has Cyclomatic Complexity value: 1" ln="26" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getArea' has Essential Complexity value: 1" ln="26" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'getArea' is both virtual and inline" ln="26" sev="3" auth="devtest" rule="OOP-25" tool="c++test" cat="OOP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Literal constant '3.141592653' is used" ln="26" sev="3" auth="devtest" rule="JSF-151" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="36" locEndLn="26" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Literal constant '3.141592653' is used" ln="26" sev="3" auth="devtest" rule="CODSTA-26" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="36" locEndLn="26" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Literal constant '3.141592653' is used" ln="26" sev="2" auth="devtest" rule="AUTOSAR-A5\_1\_1-a" tool="c++test" cat="AUTOSAR-A5\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="36" locEndLn="26" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Literal constant '3.141592653' is used" ln="26" sev="3" auth="devtest" rule="HICPP-5\_1\_1-a" tool="c++test" cat="HICPP-5\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="36" locEndLn="26" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Make virtual functions nonpublic, and public functions nonvirtual: getArea" ln="26" sev="3" auth="devtest" rule="CODSTA-CPP-25" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: getArea" ln="26" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="26" sev="3" auth="devtest" rule="FORMAT-43" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="0" locEndLn="26" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' braces are not placed in the same column" ln="26" sev="3" auth="devtest" rule="JSF-060\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="0" locEndLn="26" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Opening '{' and closing '}' should be in the same column" ln="26" sev="3" auth="devtest" rule="FORMAT-34" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="0" locEndLn="26" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="26" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="0" locEndLn="26" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="26" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="69" locEndLn="26" locEndPos="70" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="26" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="69" locEndLn="26" locEndPos="70" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="26" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="69" locEndLn="26" locEndPos="70" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the closing brace '}' on its own line" ln="26" sev="3" auth="devtest" rule="FORMAT-03" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="69" locEndLn="26" locEndPos="70" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="26" sev="3" auth="devtest" rule="JSF-061" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="0" locEndLn="26" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="26" sev="3" auth="devtest" rule="FORMAT-42" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="0" locEndLn="26" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="26" sev="3" auth="devtest" rule="JSF-060\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="0" locEndLn="26" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put the opening brace '{' on its own line" ln="26" sev="3" auth="devtest" rule="FORMAT-02" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="0" locEndLn="26" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Return type is not placed in line before function 'getArea'" ln="26" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be declared with the 'override' specifier" ln="26" sev="3" auth="devtest" rule="CODSTA-MCPP-05" tool="c++test" cat="CODSTA-MCPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be declared with the 'override' specifier" ln="26" sev="2" auth="devtest" rule="AUTOSAR-A10\_3\_2-a" tool="c++test" cat="AUTOSAR-A10\_3\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be declared with the 'override' specifier" ln="26" sev="3" auth="devtest" rule="HICPP-10\_2\_1-a" tool="c++test" cat="HICPP-10\_2\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@brief' tag" ln="26" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@brief' tag" ln="26" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@return' tag" ln="26" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'getArea' function should be preceded by a comment that contains the '@return' tag" ln="26" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="26" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="4" locEndLn="26" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="26" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="4" locEndLn="26" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="26" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="4" locEndLn="26" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="26" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="4" locEndLn="26" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="26" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="4" locEndLn="26" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="26" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="4" locEndLn="26" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="26" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="4" locEndLn="26" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'double' should not be used" ln="26" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="4" locEndLn="26" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'getArea' function is not preceded by a comment" ln="26" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'getArea' function is not preceded by a comment" ln="26" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The name 'getArea' should be composed only of lowercase letters" ln="26" sev="3" auth="devtest" rule="JSF-051" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The name 'getArea' should be composed only of lowercase letters" ln="26" sev="3" auth="devtest" rule="NAMING-44" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use the virtual keyword for 'getArea' function" ln="26" sev="3" auth="devtest" rule="OOP-21" tool="c++test" cat="OOP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use the virtual keyword for 'getArea' function" ln="26" sev="2" auth="devtest" rule="MISRA2008-10\_3\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="11" locEndLn="26" locEndPos="12" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="26" sev="5" auth="devtest" rule="MISRA2008-0\_4\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="36" locEndLn="26" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="26" sev="3" auth="devtest" rule="OWASP2019-API9-d" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="36" locEndLn="26" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="26" sev="3" auth="devtest" rule="COMMENT-10" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="36" locEndLn="26" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="26" sev="2" auth="devtest" rule="AUTOSAR-M0\_4\_2-a" tool="c++test" cat="AUTOSAR-M0\_4\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="26" locStartPos="36" locEndLn="26" locEndPos="37" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="'contains' shall be declared as unsigned int or signed int" ln="27" sev="3" auth="devtest" rule="PORT-13" tool="c++test" cat="PORT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'contains' has Cyclomatic Complexity value: 1" ln="27" sev="5" auth="devtest" rule="METRICS-29" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'contains' has Essential Complexity value: 1" ln="27" sev="5" auth="devtest" rule="METRICS-33" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="27" sev="2" auth="devtest" rule="MISRA2008-2\_10\_1" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="27" sev="3" auth="devtest" rule="HICPP-2\_4\_1-a" tool="c++test" cat="HICPP-2\_4\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="27" sev="2" auth="devtest" rule="AUTOSAR-M2\_10\_1-a" tool="c++test" cat="AUTOSAR-M2\_10\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Identifier name: 'point' differs only by case from its type name: 'Point'" ln="27" sev="3" auth="devtest" rule="NAMING-47" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Naming convention not followed: contains" ln="27" sev="3" auth="devtest" rule="NAMING-17" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Parameter 'point' is not validated before use" ln="27" sev="3" auth="devtest" rule="CERT\_C-API00-a" tool="c++test" cat="CERT\_C-API00" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Parameter 'point' is not validated before use" ln="27" sev="3" auth="devtest" rule="CODSTA-86" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Return type is not placed in line before function 'contains'" ln="27" sev="3" auth="devtest" rule="FORMAT-28" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'contains' function should be preceded by a comment that contains the '@brief' tag" ln="27" sev="3" auth="devtest" rule="COMMENT-14" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'contains' function should be preceded by a comment that contains the '@brief' tag" ln="27" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-a" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'contains' function should be preceded by a comment that contains the '@return' tag" ln="27" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'contains' function should be preceded by a comment that contains the '@return' tag" ln="27" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'point' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="27" sev="3" auth="devtest" rule="COMMENT-14\_b" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The 'point' parameter does not have a corresponding '@param' tag in the comment before the function declaration" ln="27" sev="2" auth="devtest" rule="AUTOSAR-A2\_7\_3-b" tool="c++test" cat="AUTOSAR-A2\_7\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="27" sev="3" auth="devtest" rule="HICPP-7\_1\_6-b" tool="c++test" cat="HICPP-7\_1\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="27" sev="4" auth="devtest" rule="MISRAC2012-DIR\_4\_6-b" tool="c++test" cat="MISRAC2012-DIR\_4\_6" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="27" sev="3" auth="devtest" rule="MISRA2004-6\_3\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="27" sev="2" auth="devtest" rule="JSF-209\_b" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="27" sev="4" auth="devtest" rule="MISRA2012-DIR-4\_6\_b" tool="c++test" cat="MISRA2012-DIR" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="27" sev="3" auth="devtest" rule="HICPP-3\_5\_1-b" tool="c++test" cat="HICPP-3\_5\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="27" sev="4" auth="devtest" rule="MISRA2008-3\_9\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The basic numerical type 'int' should not be used" ln="27" sev="3" auth="devtest" rule="MISRA-013" tool="c++test" cat="MISRA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'contains' function is not preceded by a comment" ln="27" sev="3" auth="devtest" rule="COMMENT-04" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The definition of the 'contains' function is not preceded by a comment" ln="27" sev="4" auth="devtest" rule="JSF-134" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="8" locEndLn="27" locEndPos="9" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'point' differs only by case from identifier 'Point' declared in file 'Point.hpp'" ln="27" sev="3" auth="devtest" rule="NAMING-45" tool="c++test" cat="NAMING" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The identifier 'point' differs only by case from identifier 'Point' declared in file 'Point.hpp'" ln="27" sev="3" auth="devtest" rule="JSF-048" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'point' of function 'contains' is passed by value" ln="27" sev="3" auth="devtest" rule="OPT-14" tool="c++test" cat="OPT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'point' of function 'contains' is passed by value" ln="27" sev="4" auth="devtest" rule="AUTOSAR-A8\_4\_3-a" tool="c++test" cat="AUTOSAR-A8\_4\_3" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'point' of function 'contains' is passed by value" ln="27" sev="4" auth="devtest" rule="JSF-125\_a" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="The parameter 'point' of function 'contains' is passed by value" ln="27" sev="4" auth="devtest" rule="JSF-117" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="23" locEndLn="27" locEndPos="24" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="27" sev="3" auth="devtest" rule="CODSTA-223" tool="c++test" cat="CODSTA" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use the fixed width integer type from &lt;cstdint> instead of the 'int' basic numerical type" ln="27" sev="2" auth="devtest" rule="AUTOSAR-A3\_9\_1-b" tool="c++test" cat="AUTOSAR-A3\_9\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="27" locStartPos="4" locEndLn="27" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Percentage of comment lines vs. all method's lines is: 0" ln="28" sev="3" auth="devtest" rule="METRICS-19" tool="c++test" cat="METRICS" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="28" locStartPos="0" locEndLn="28" locEndPos="1" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="'return' statement should be used with parenthesis" ln="29" sev="3" auth="devtest" rule="FORMAT-25\_b" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="5" locEndLn="29" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Function 'contains' has declared return type 'int' but returns value of type 'bool'" ln="29" sev="3" auth="devtest" rule="PB-05" tool="c++test" cat="PB" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="5" locEndLn="29" locEndPos="6" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Implicit conversion from integral to floating type in comparison expression" ln="29" sev="2" auth="devtest" rule="MISRA2008-5\_0\_5\_b" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="12" locEndLn="29" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Implicit conversion from integral to floating type in comparison expression" ln="29" sev="3" auth="devtest" rule="CERT\_C-EXP39-a" tool="c++test" cat="CERT\_C-EXP39" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="12" locEndLn="29" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Implicit conversion from integral to floating type in comparison expression" ln="29" sev="2" auth="devtest" rule="AUTOSAR-M5\_0\_5-a" tool="c++test" cat="AUTOSAR-M5\_0\_5" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="12" locEndLn="29" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Implicit conversion from integral to floating type in comparison expression" ln="29" sev="3" auth="devtest" rule="MISRA2004-10\_1\_b" tool="c++test" cat="MISRA2004" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="12" locEndLn="29" locEndPos="13" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="29" sev="4" auth="devtest" rule="JSF-043" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="4" locEndLn="29" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="29" sev="5" auth="devtest" rule="FORMAT-01" tool="c++test" cat="FORMAT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="4" locEndLn="29" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Non-ascii tab found" ln="29" sev="5" auth="devtest" rule="HICPP-2\_1\_1-a" tool="c++test" cat="HICPP-2\_1\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="4" locEndLn="29" locEndPos="5" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Put function 'getPosition' that are used in the body of inline member function into separate file" ln="29" sev="3" auth="devtest" rule="CODSTA-CPP-15" tool="c++test" cat="CODSTA-CPP" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="35" locEndLn="29" locEndPos="36" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="29" sev="5" auth="devtest" rule="MISRA2008-0\_4\_2" tool="c++test" cat="MISRA2008" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="54" locEndLn="29" locEndPos="55" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="29" sev="3" auth="devtest" rule="OWASP2019-API9-d" tool="c++test" cat="OWASP2019-API9" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="54" locEndLn="29" locEndPos="55" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="29" sev="3" auth="devtest" rule="COMMENT-10" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="54" locEndLn="29" locEndPos="55" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="floating-point arithmetic is not documented" ln="29" sev="2" auth="devtest" rule="AUTOSAR-M0\_4\_2-a" tool="c++test" cat="AUTOSAR-M0\_4\_2" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="29" locStartPos="54" locEndLn="29" locEndPos="55" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use // comments only" ln="33" sev="3" auth="devtest" rule="HICPP-2\_3\_1-a" tool="c++test" cat="HICPP-2\_3\_1" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="33" locStartPos="7" locEndLn="33" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use // comments only" ln="33" sev="2" auth="devtest" rule="JSF-126" tool="c++test" cat="JSF" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="33" locStartPos="7" locEndLn="33" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

<StdViol msg="Use // comments only" ln="33" sev="3" auth="devtest" rule="COMMENT-01" tool="c++test" cat="COMMENT" lang="cpp" locType="sr" config="1" hash="1537905639" locStartln="33" locStartPos="7" locEndLn="33" locEndPos="8" locFile="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp"/>

</StdViols>

<Supps/>

<TestedFilesDetails>

<Total name="Suppressed / Total" supp="0" total="3236">

<Project name="cicd.findings.cpptest.professional.static.analysis.report" supp="0" total="3236">

<Res name="cicd.findings.cpptest.professional.static.analysis.report" supp="0" total="3236">

<Res loc="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Point.hpp" locType="sr" name="Point.hpp" supp="0" total="262"/>

<Res loc="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DivisionByZero.cpp" locType="sr" name="DivisionByZero.cpp" supp="0" total="109"/>

<Res loc="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/DeadLock.cpp" locType="sr" name="DeadLock.cpp" supp="0" total="1868"/>

<Res loc="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/NullPointer.cpp" locType="sr" name="NullPointer.cpp" supp="0" total="178"/>

<Res loc="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/MemoryLeak.cpp" locType="sr" name="MemoryLeak.cpp" supp="0" total="324"/>

<Res loc="/cicd.findings.cpptest.professional.static.analysis.report/cicd.findings.cpptest.professional.static.analysis.report/Shapes.hpp" locType="sr" name="Shapes.hpp" supp="0" total="495"/>

</Res>

</Project>

</Total>

</TestedFilesDetails>

</CodingStandards>

<Metrics ownerId="com.parasoft.xtest.checkers.api.standards" time="0:01:19">

</Metrics>

<AssocUrls/>

</ResultsSession>