Code Review - MISRA 2012 rules

D is set for Decidable, U for Undecidable.

| **Code review reference** | **Type** | **D/U** | **Description** | **Level** |
| --- | --- | --- | --- | --- |
| M1.1 | Error | D | ANSI C error: <name> | Required |
| M1.1W | **Error** | D | ANSI C warning: <name> | Required |
| M1.2 | Error | U | Use of #pragma <name> should always be encapsulated and documented | Advisory |
| E1.1 | Error | D | Function max number of line | Required |
| E.1.2 | Error | D | Function max V(g) | Required |
| M2.1 | Error | U | a project shall not contain unreachable code | Required |
| M2.2.1 | Error | U | A non-null statement should either have a side effect or change the control flow | Required |
| **M2.2.2** | Error | U | The function <name> is never referenced | Required |
| M2.3 | Warning | D | Type <name> is never used | Advisory |
| M2.4 | Warning | D | Tag <name> is never used | Advisory |
| M2.5 | Warning | D | Macro <name> is never used | Advisory |
| M2.6 | Warning | D | A function should not contain unused label declarations | Advisory |
| M2.7 | Warning | D | There should be no unused parameters in functions | Advisory |
| M3.1.1 | Error | D | The character sequence /\* should not be used within a comment | Required |
| M3.1.2 | Error | D | The character sequence // should not be used within a 'C-style' comment | Required |
| M3.2 | Error | D | Line-splicing shall not be used in // comments | Required |
| E3.1 | Error | D | A null statement in original source code should be on a separate line and the semicolon should be followed by at least one white space and then a comment | Required |
| M4.1 | Error | D | Octal and hexadecimal escape sequences shall be terminated | Required |
| M4.2 | Warning | D | Trigraphs should not be used | Advisory |
| E4.1 | Error | D | Only ISO C escape sequences are allowed | Advisory |
| E.4.2 | Error | D | Only ISO C escape sequences are allowed(\v) | Advisory |
| M5.1.1 | Error | D | External identifiers shall be distinct in the first 31 characters | Required |
| M5.1.2 | Error | D | External identifiers shall be distinct in the first 6 characters ignoring case | Required |
| M5.2 | Error | D | Identifiers <name>declared in the same scope and name space shall be distinct. Identifier identical in the first <param> characters already found in <location> | Required |
| M5.3 | Error | D | Identifier <name> declared in an inner scope shall not hide an identifier declared in an outer scope. Identifier identical in the first <param> characters already found in <location> | Required |
| M5.4.1 | Error | D | Macros '%name%' and '%name%' are identical in the first '%param%' characters | Required |
| M5.4.2 | Error | D | Macros '%name%' and '%name%' are identical in the first '%param%' characters ignoring case. | Required |
| M5.5.1 | Error | D | Macro '%name%' and identifier '%name%' are identical in the first '%param%' characters. | Required |
| M5.5.2 | Error | D | Macro '%name%' and identifier '%name%' are identical in the first '%param%' characters ignoring case. | Required |
| M5.6 | Error | D | Macro '%name%' and identifier '%name%' are identical in the first '%param%' characters ignoring case.The typedef name '%name%' should not be reused except for its tag. Name already found in %location% | Required |
| M5.7.1 | Error | D | The tag name '%name%' should not be reused | Required |
| M5.7.2 | Error | D | A struct and union cannot use the same tag name | Required |
| M5.8 | Error | D | Identifiers that define objects or functions with external linkage shall be unique | Required |
| M5.9 | Error | D | Identifiers that define objects or functions with internal linkage should be unique | Advisory |
| E5.1 | Error | D | External identifiers shall not be ambiguous because of possible character confusion. | Advisory |
| E5.2 | Error | D | External identifiers shall not be ambiguous because of character repetition | Advisory |
| E5.3 | Warning | D | The identifier '%name%' should not be reused. Identifier already found in %location% | Advisory |
| E5.4 | Error | D | Identifier '%name%' in an inner scope hides the same identifier in an outer scope : %location% | Advisory |
| E5.5 | Error | D | The typedef name '%name%' should not be reused even for its tag. Name already found in %location% | Advisory |
| M6.1.1 | Error | D | Bit fields should only be of type 'unsigned int' or 'signed int' | Required |
| M6.1.2 | Error | D | Bit fields should not be of type 'enum' | Required |
| M6.1.3 | Error | D | Bit fields should only be of explicitly signed or unsigned type | Required |
| M6.1.4 | Error | D | Bit fields should not be of type 'bool' under c99 | Required |
| M6.1.5 | Error | D | Bit fields should not be of type 'boolean' outside c99 | Required |
| M6.2 | Error | D | Single-bit fields shall not be of a signed type | Required |
| E6.1 | Warning | D | The C language numeric type '%name%' should not be used directly but instead used to define typedef | Required |
| E6.2 | Warning | D | The implicit 'int' type should not be used | Required |
| M7.1 | Error | D | octal constans shall not be used | Required |
| M7.2 | Error | D | a u" or "U" suffix shall be applied to all integer constants that are represented in an unsigned type" | Required |
| M7.3 | Error | D | the lowercase characted l" shall not be used in a literal suffix" | Required |
| M7.4 | Error | D | a string litteral shall not be assigned to an object unless the object's type is pointer to a const-qualified char | Required |
| M8.1 | Error | D | types shall be explicitly specified | Required |
| M8.2.1 | Error | D | The function prototype should name all its parameters | Required |
| M8.2.2 | Error | D | Functions with no parameters should use the void type | Required |
| M8.2.3 | Error | D | The type of parameter '%name%' should be explicitly stated | Required |
| M8.3.1 | Error | D | Parameters and return types should use compatible type in the declaration and in the definition | Required |
| M8.3.2 | Error | D | The identifiers used in the prototype and definition should be the same | Required |
| M8.4.1 | Error | D | A prototype for the global function '%name%' should be declared before defining the function | Required |
| M8.4.2 | Error | D | A prototype for the global object '%name%' should be declared before defining the object | Required |
| M8.4.3 | Error | D | If objects or functions are declared multiple times their types should be compatible | Required |
| M8.5 | Error | D | Identifiers '%name%' that declare objects or functions with external linkage shall be declared once in one and only one file | Required |
| M8.6 | Error | D | Identifiers '%name%' that declare objects or functions with external linkage shall be unique | Required |
| **M8.7.1** | Warning | D | Global object '%name%' that is only used within the same file should be declared using the static storage-class specifier. | Advisory |
| **M8.7.12** | Warning | D | Global function '%name%' that are only used within the same file should be declared using the static storage-class specifier. | Advisory |
| M8.8 | Error | D | The static storage class specifier shall be used in all declarations of objects and functions that have internal linkage | Required |
| M8.9 | Warning | D | An object should be defined at block scope if its identifier only appears in a single function | Advisory |
| M8.10 | Error | D | Inline function '%name%' should be static | Required |
| M8.11 | Warning | D | when an array with external linkage is declared, its size should be explicitly specified | Advisory |
| M8.12 | Error | D | Enumeration member '%name%' have a not unique implicitly-specified value | Required |
| M8.14 | Error | D | the restrict type qualifier shall not be used | Required |
| E.8.1 | Error | D | Parameters and return types should use exactly the same type names in the declaration and in the definition | Required |
| E.8.2 | Error | D | A prototype for the static function '%name%' should be declared before defining the function | Required |
| E.8.3 | Error | D | Static function '%name%' should only be declared in a single file. Redundant declaration found at: %location% | Required |
| E.8.4 | Error | D | Static object '%name%' should only be declared in a single file. Redundant declaration found at: %location% | Required |
| E.8.5 | Error | D | Either all members or only the first member of an enumerator list should be initialized | Required |
| E.8.6 | Error | D | The body of function '%name%' should not be located in a header file | Required |
| E.8.7 | Error | D | The memory storage (definition) for the variable '%name%' should not be in a header file | Required |
| E.8.8 | Error | D | Functions should not be declared at block scope | Required |
| **E.8.9** | Error | D | The global object or function '%name%' should have exactly one external definition. Redundant definition found in %location% | Required |
| **E.8.10** | Error | D | The global object or function '%name%' should have exactly one external definition. No definition found | Required |
| E.8.11 | Error | D | Use the const qualification for variable '%name%' which is pointer and which is not used to change the pointed object | Required |
| **E.8.12** | Warning | D | The object '%name%' is never referenced | Required |
| M9.2 | Error | D | the initializer for an aggregate or union shall be enclosed in braces | Required  Exception not covered |
| M9.3 | w | D | arrays shall not be partially initialized | Required  Exception not covered |
| E9.1 | Error | D | Variables with automatic storage duration should be initialized before being used | Required |
| E9.2 | Error | D | The global variable '%name%' is not initialized | Required |
| M10.1.1 | Error | D | Constraint violation : can't use floating type as operand of "[], %, &lt;&lt;, >>, ~, &amp;, |, ^" | Required |
| M10.1.2 | Error | D | Operand should be boolean | Required |
| M10.1.3 | Error | D | Can't use a boolean as a numeric value | Required |
| M10.1.4 | Error | D | Can't use a char as a numeric value | Required |
| M10.1.5 | Error | D | Can't use a not anonymous enum as a numeric value | Required |
| M10.1.6 | Error | D | Shift and bitwise operations should be performed on unsigned value | Required |
| M10.1.7 | Error | D | Right hand operand of shift operation should be performed on unsigned value | Required |
| M10.1.8 | Error | D | Unary minus operation should not be performed on unsigned value | Required |
| M10.2 | Error | D | Expressions of essentially character type shall not be used inappropriately in addition and substraction operations | Required |
| M10.3.1 | Error | D | The value of an expression shall not be assigned to an object with a narrower essential type | Required |
| M10.3.2 | Error | D | The value of an expression shall not be assigned to an object with a different essential type category | Required |
| M10.4 | Error | D | both operands of an operator in which the usual arithmetic conversions are performed shall have the same essential type category | Required |
| M10.5 | Warning | D | the value of an expression should not be cast to an inapropriate essential type | Advisory |
| M10.6 | Error | D | The value of a composite expression shall not be assigned to an object with wider essential type | Required |
| M10.7 | Error | D | if a composite expression is used as one operand of an operation in which the usual arithmetic conversions are performed then the other operand shall not have wider essential type | Required |
| M10.8 | Error | D | the value of a composite expression shall not be cast to a different essential type category or a wider essential type | Required |
| E10.1 | Error | D | When using operator '~' or '&amp;lt;&amp;lt;' on 'unsigned char' or 'unsigned int', you should always cast returned value | Required |
| M11.1 | Error | D | A function pointer should not be converted to another type of pointer | Required |
| M11.2 | Error |  | conversions shall not be performed between a pointer to an incomplete type and any other type | Required |
| M11.3.1 | Error |  | Casting an object pointer type to a different object pointer type should not occur | Required |
| M11.3.2 | Error |  | Casting an object pointer type to a different object pointer type should not occur, especially when object sizes are not the same | Required |
| M11.3.3 | Error |  | An object pointer should not be converted to another type of pointer | Required |
| M11.4 | Warning |  | Casting a pointer type to an integer type should not occur | Advisory |
| M11.5 | Warning |  | a conversion should not be performed from pointer to void into pointer to object | Advisory |
| M11.6 | Error |  | a cast shall not be performed between pointer to void and and an arithmetic type | Required |
| M11.7 | Error |  | a cast shall not be performed between pointer to object and a non-integer arithmetic type | Required |
| M11.8 | Error |  | Casting of pointers to a type that removes any const or volatile qualification on the pointed object should not occur | Required |
| M12.1.1 | warning |  | Implicit operator precedence may cause ambiguity. Use parenthesis to clarify this expression | Advisory |
| M12.1.2 | warning |  | Implicit bitwise operator precedence may cause ambiguity. Use parenthesis to clarify this expression | Advisory |
| M12.1.3 | warning |  | Parenthesis should be used around expressions that are operands of a logical &amp;amp;&amp;amp; or || | Advisory |
| M12.3 | warning |  | the comma operator should not be used | Advisory |
| E12.1 | warning |  | The operator on a Boolean expression should be a logical operator ( &amp;&amp;, || or !) | Advisory |
| E12.2 | warning |  | Ternary expression '?:' should not be used | Advisory |
| E12.3 | error |  | Expressions should not cause a side effect assignment | Advisory |
| E12.4 | error |  | The equal or not equal operator should not be used in floating-point expressions | Advisory |
| M13.3 | Warning |  | 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 | Advisory |
| M13.4.1 | Warning |  | Boolean expressions should not contain assignment operators. | Advisory |
| M13.4.2 | Warning |  | The result of an assignment operator should not be used in an expression | Advisory |
| M13.6 | Error |  | the operand of the sizeof operator shall not contain any expression which has potential side effects | Required |
| E13.1 | Error |  | Boolean expressions should not contain side effect operators | Required |
| E13.2 | Error |  | An expression that contains a side effect should not be used in the right-hand operand of a logical &amp;&amp; or || operator | Required |
| E13.3 | Error |  | The function in the right-hand operand of a logical && or || operator might cause side effects | Required |
| M14.1.1 | Error |  | Floating-point variables should not be used to control a for statement | Required |
| M14.2.1 | Error |  | Only loop counter should be initialized in a for loop initialization part | Required |
| M14.2.2 | Error |  | In the 'update part' of a 'for statement', only 'loop counter' should be updated | Required |
| M14.2.3 | Error |  | There should be one and only one loop counter for loop statement | Required |
| M14.2.4 | Error |  | Loop counter of a 'for statement' should not be modified within the body of the loop | Required |
| M14.3.1 | Error |  | Invariant Boolean expressions should not be used | Required |
| M14.4 | Error |  | Non-Boolean values that are tested against zero should have an explicit test | Required |
| M15.1 | Warning |  | the goto statement should not be used | Advisory |
| M15.2 | Error |  | the goto statement shall jump to a label declared later in the same function | Required |
| M15.3 | Error |  | any label referenced by a goto statement shall be declared in the same block, or in any block enclosing the goto statement | Required |
| M15.4 | Warning |  | There should be no more than one break or goto statement used to terminate any iteration statement | Advisory |
| M15.5 | Warning |  | a function should have a single point of exit at the end | Advisory |
| M15.6.1 | Error |  | The switch statement should be followed by a compound statement | Required |
| M15.6.1 | Error |  | The switch statement should be followed by a compound statement | Required |
| M15.6.2 | Error |  | The while statement should be followed by a compound statement | Required |
| M15.6.3 | Error |  | The do..while statement should contain a compound statement | Required |
| M15.6.4 | Error |  | The for statement should be followed by a compound statement | Required |
| M15.6.5 | Error |  | The if (expression) construct should be followed by a compound statement | Required |
| M15.6.6 | Error |  | The else keyword should be followed by a compound statement | Required |
| M15.7 | Error |  | all if ... else constructs shall be terminated with an else statement | Required |
| E15.1 | Error |  | Do not use the continue statement | Required |
| E15.2 | Error |  | Only one break statement should be used within a loop | Required |
| E15.3 | Error |  | The return keyword should not be used in a conditional block | Required |
| E15.4 | Error |  | The else keyword should be followed by either a compound statement or another if statement. | Required |
| M16.1 | Error |  | all switch statement should be well formed | Required |
| M16.2 | Error |  | a switch label shall only be used when the most closely-enclosing copound statement is the body of a switch statement | Required |
| M16.3 | Error |  | an unconditional break statement shall terminate every switch-clause | Required |
| M16.4 | Error |  | every switch statement shall have a default label | Required |
| M16.5 | Error |  | a default label appear as either the first or the last switch label of a switch statement | Required |
| M16.6 | Error |  | every switch statement shall have at least two switch-clauses | Required |
| M16.7 | Error |  | a switch expression shall not have essentially Boolean type | Required |
| E16.1 | Error |  | Case char value is applicable only if the switch statement value is plain character variable | Required |
| E16.2 | Error |  | A constant should not be used as a switch expression | Required |
| E16.3 | Error |  | The switch expression should not have side effects | Required |
| M17.1.1 | Error |  | The function '%name%' should not have a variable number of arguments | Required |
| M17.1.2 | Error |  | The va\_list, va\_arg, va\_start, va\_end and va\_copy functions of &lt;stdarg.h> shall not be used | Required |
| M17.2.1 | Error |  | Recursive functions are not allowed. The function '%name%' is directly recursive | Required |
| M17.2.2 | Error |  | Recursive functions are not allowed. The function '%name%' is recursive when calling '%name%' | Required |
| M17.3 | Error |  | a function shall not be declared implicitly | Required |
| M17.4 | Error |  | all exit paths from a function with non-void return type shall have an explicit return statement with an expression | Required |
| M17.6 | Error |  | the declaration of an array parameter shall not contain the static keyword between the [] | Advisory |
| M17.7 | Error |  | the value returned by function having non-void return type shall be used | Required |
| E17.1 | Error |  | The number of arguments used in the call does not match the number declared in the prototype | Advisory |
| E17.2 | Error |  | Use the const qualification for parameter '%name%' which is pointer and which is not used to change the pointed object | Advisory |
| E17.3 | Error |  | Function identifiers should always use a parenthesis or a preceding &amp; | Advisory |
| M18.4 | Error |  | the +, -, += and -= operators should not be applied to an expression of pointer type | Advisory |
| M18.5 | Error |  | declarations should contain no more than two levels of pointer nesting | Advisory |
| M18.7 | Error |  | flexible arrays members shall not be declared | Required |
| M18.8 | Error |  | variable-length array types shall not be used | Required |
| M19.2 | Warning |  | the union keyword should not be used | Advisory |
| E19.1 | Error |  | Structure or union types '%name%' should be finalized before the end of the compilation units | Advisory |
| M20.1 | Warning |  | #include directive should only preceded by preprocessor directives or comments | Advisory |
| M20.2 | Error |  | the ', or \ character and the /\* or // character sequences shall not occur in a header file name" | Required |
| M20.3 | Error |  | the #include directive shall be followed by either a &lt;filename> or a filename" sequence" | Required |
| M20.4 | Error |  | a macro shall not be defined with the same name as a keyword | Required |
| M20.5 | Warning |  | #undef should not be used | Advisory |
| M20.6 | Error |  | token that look like a preprocessing directive should not occur withing a macro argument | Required |
| M20.7 | Error |  | expressions resulting from the expansion of macro parameters shall be enclosed in parenthesis | Required |
| M20.8 | Error |  | the controlling expression of a #if or #elif preprocessing directive shall evaluate to 0 or 1 | Required |
| M20.9 | Error |  | all identifiers used in the controlling expression of #if or #elif preprocessing directives shall be #define'd before evaluation | Required |
| M20.10 | Warning |  | the # and ## preprocessor operators should not be used | Advisory |
| M20.11 | Error |  | A macro parameter immediately following a # operator shall not immediately be followed by a ## operator | Required |
| M20.12 | Error |  | a macro parameter used as an operand to the # and ## operators shall only be used as an operand to these operators | Required |
| M20.13 | Error |  | a line whose first token is # shall be a valid preprocessing directive | Required |
| M20.14 | Error | Error | 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 | Required |
| E20.1 | Error |  | Header file contents should be protected against multiple inclusions | Required |
| E20.2 | Error |  | The # or ## preprocessor operator should not be used more than once | Required |
| E20.3 | Error |  | Missing argument when calling the macro | Required |
| E20.4 | Error |  | Only use the 'defined' preprocessor operator with a single identifier | Required |
| E20.5 | Error |  | Macro definitions or '#undef' should not be located within a block | Required |
| E20.6 | Error |  | A C macro should only be expanded to a constant, a braced initialiser, a parenthesised expression, a storage class keyword, a type qualifier, or a do-while-zero block | Required |
| M21.1.1 | Error |  | #define and #undef shall not be used on a reserved identifier or reserved macro name: Identifier %name% already found in &lt;%libname%> | Required |
| M21.1.2 | Error |  | #define and #undef shall not be used on identifier beginning with an underscore or on 'defined' keyword | Required |
| M21.2.1 | Error |  | Declared identifier should not be a reserved identifier or reserved macro name: Identifier %name% already found in <%libname%> | Required |
| M21.2.2 | Error |  | Declared identifier should not begin with an underscore or be 'defined' keyword | Required |
| M21.3 | Error |  | the memory allocation and deallocation functions of &lt;stdlib.h> shall not be used | Required |
| M21.4 | Error |  | the standard header file &lt;setjmp.h> shall not be used | Required |
| M21.5 | Error |  | the standard header file &lt;signal.h> shall not be used | Required |
| M21.6.1 | Error |  | The input/output library &lt;stdio.h> shall not be used in production code | Required |
| M21.6.2 | Error |  | The input/output library &lt;wchar.h> shall not be used in production code | Required |
| M21.7 | Error |  | the library macro or functions atof, atoi, atol and atoll of &lt;stdlib.h> shall not be used | Required |
| M21.8 | Error |  | the library macro or functions abort, exit, getenv and system of &lt;stdlib.h> shall not be used | Required |
| M21.9 | Error |  | the library macro or functions bsearch and qsort of &lt;stdlib.h> shall not be used | Required |
| M21.10 | Error |  | the standard library time and date functions shall not be used | Required |
| M21.11 | Error |  | the standard header file &lt;tgmath.h> shall not be used | Required |
| M21.12 | Warning |  | The library macro or function 'feclearexcept, fegetexceptflag, feraiseexcept, fesetexceptflag, fetestexcept, FE\_INEXACT, FE\_DIVBYZERO, FE\_UNDERFLOW, FE\_OVERFLOW, FE\_INVALID or FE\_ALL\_EXCEPT' should not be used. | Advisory |
| E21.1 | Error |  | The variable 'errno' should not be used | Required |
| E21.2 | Error |  | The macro 'offsetof' should not be used | Required |
| E21.3 | Error |  | The library macro or function 'setjmp,longjmp,sigsetjmp,siglongjmp' should not be used | Required |
| Rule U99.1 | Error |  | User custom rule |  |