Treyton Davis

CS-405

June 7, 2024

Module 5-3 Assignment

A screenshot of a computer

Description automatically generated

For Visual Studio debug errors, it said that abort() was called due to the assertion failing to verify that my\_function() was returned with 3. The reason for this error was due to my\_function() being listed as bool instead of int to allow for an integer to be returned.

A screenshot of a computer

Description automatically generated

There are a few different warnings noticed by CppCheck that was not detected by Visual Studio. The one highlighted in the screenshot says that there was a variable being assigned inside an assert command, which could be a bug due to asserts verifying the variable is correct. That would be a risk due to potential of changing the variable while the program is running to allow for data leaks to occur. The next warning says the same about my\_function() even though the code is written correctly, we shouldn’t be calling a function inside an assert. The next warning is about the variable tok inside the foo class. CppCheck says there is no effect outside of function, since all it does is change tok to the next variable inside tok. This is not a risk due to the code not being used in front of the user. The first error is talking about a function parameter being assigned the address of a variable instead of the value of the variable. This can be a risk due to a user potentially having access to see the variables. The next one is about the assert(my\_function() == 3). This was discovered by Visual Studio while trying to run the program. This can be a risk due to causing an abort() to occur when trying to open the program. The next warning is about an out of bounds error. It shows that if count is equal to 1000, then assign buf at count to equal 0. This is a risk due to potential data loss and data being overwritten. The next warning is about a variable not being initialized in the copy constructor. This is not a risk because it is in a class, not a function so it should not be user-accessible. The next warning is about tok again, this time being a potential null pointer dereference. This could be a risk due to potentially referencing NULL while trying to find the next variable. The next error is about an exception thrown in a function declared to not throw an exception. This is a bug, not a risk due to just throwing an exception that shouldn’t be there. The final error is about using iterator for local container that may be invalid. This could be a risk since iterator could potentially erase all data inside the items container.