Module Five Static Analysis

CS 405

Miranda Putnam

**Screenshots**

Errors after build (Initial Xcode errors):

**A screenshot of a computer

Description automatically generated**

Errors after analysis (Clang errors):

**A screenshot of a computer

Description automatically generated**

**Summary**

When I first built the provided code, Xcode flagged semantic and unused entity issues. The first issue that Xcode caught was that the function DontThrow has a non-throwing exception, but still can throw. The next issue was that the result of a comparison with a constant to a Boolean expression is always false. Then, there were unused variables defined within the code. I then ran the Clang analysis, which found further issues within the code. These included a dead store error and logic errors. The first error was that the value stored to ‘tok’ is never read by the program. The next error was that the called pointer in the program at lines 108 and 109 is null. This is a while loop that assumes the pointer value is null, the loop executes zero times, and the called pointer is null. This logically does not make sense. While both Xcode and Clang identified important issues, Clang was able to find issues that have a greater effect on the performance of the program. The issues within the while loop that Clang identified are important for a programmer to be aware of, because a while loop that does not logically make sense, and ends up not working will affect the outcome of the program. Because of this, the issues identified by Clang are a risk to the program. This while loop could be fixed by making ‘tok’ not null. This would allow the loop to execute and read a value into ‘tok’. The issues identified by Xcode are important for a programmer to be aware of as well, but most are not high risk and will not directly affect the outcome of the program. For example, unused variables won’t prevent the program from running, however, it’s not best practice to have code that is not relevant to the outcome of the program. More serious issues that Xcode identified are the semantics issues. A programmer should not allow a non-throwing expression to throw, and should not compare a Boolean expression with a constant, as the result is always false. Overall, both Xcode and Clang identified valuable issues within the provided code, each with a different risk level, but all worth noting and fixing by the programmer.