Daniel Coffey

4-1 Activity: Exceptions

CS 405 Secure Coding

02/06/2021

For this assignment I was tasked with the duty of being a senior software developer on a team of software developers who are responsible for a large banking web application. There was an issue of code terminating abruptly, which would then crash to the desktop without any warnings or information displayed to the user. The suspected code was found, but instead of changing the code, the decision was made to display the relative errors to the user instead of just crashing to the desktop. I was tasked with fixing this by utilizing exceptions. I needed to create an exception try-block, which will wrap code that can potentially throw exceptional errors, catch an exception derived from std::exception, catch all unhandled exceptions, throw a standard C++ exception, and create, throw, and catch a custom exception.

Exception handling has many advantages over traditional error handling. Exception handling separates error handling code from normal code. In traditional code, error handling is done utilizing if/else conditions, while with exception handling you handle errors with try/catch blocks. This separates the code for error handling from the normal flow as if/else conditionals can be blended in with the normal flow of code. It is also good to utilize exception handling because functions/methods can handle any exceptions that they choose. What this means is that a function can throw many exceptions but choose to handle only some of them. The remaining exceptions that are thrown but not caught may be handled by the caller. There is a special catch block known as the catch-all block. This catch-all handler can be used to catch all types of exceptions. This may seem like a good idea, but after doing research, I don’t think I would always utilize this. There are a few reasons for this. If you don’t know why an exception happened, you might not know how to deal with it. Some exceptions require unique attention and care to be properly handled. Exceptions should only be caught when you know exactly what to do with it and how to handle it. From my research, catching exceptions is ideal at the top level of your project where you can log it and report back to your team.

For this assignment, I didn’t really encounter any issues, I followed the “TODOs”, and it seemed straightforward. My code is commented on my thought process and steps taken along the way.

