Edwin Jones

CS-405: Secure Coding

June 1, 2025

Process Summary

In this assignment, I implemented exception handling to ensure the program does not crash abruptly. I added exception types including standard exceptions (std::runtime\_error, std::logic\_error), a custom exception (CustomException), and corresponding try-catch handlers.

I began by validating division with zero and ensured that std::runtime\_error is properly thrown and caught. I then implemented logic in do\_custom\_application\_logic() that throws a std::logic\_error and wraps it in a try-catch block. Lastly, I created and threw a custom exception and handled it explicitly in main(), followed by a general std::exception handler and a catch-all handler.

One challenge was ensuring exception types were distinct and matched to the appropriate catch blocks to demonstrate all required behaviors. I resolved this by nesting function calls and ensuring exception order followed the rubric. The code now provides robust error handling, consistent with best practices for secure and maintainable C++.

A screen shot of a computer

AI-generated content may be incorrect.