Module Four Exceptions

Jordan Whitney

CS-405 Secure Coding

Professor: Alan Spencer

7/25/2025

The testing exceptions that I used would start with a standard exception, divide by zero: thrown using “std::invalid\_argument” in “divide()”, caught in do\_division(). Another standard exception for custom logic failure “do\_even\_more\_custom\_application\_logic()” throws a “std::runtime\_error”, caught in “do\_custom\_applicationn\_logic()”. A custom exception, defined as “CustomException”(derived from std::exception), thrown from do\_custom\_application\_logic(): and caught in main. To catch any uncaught or unexpected exceptions a catch-all handler is present in main(). As well as all handlers display the exception message using “what()”, and the program completes without crahing. I wanted to isolate each function in its own try-catch block, also using specific exception types to clarify exception cause like, “std:invalid\_argument”, and “std::runtime\_error”.

A screenshot of a computer

AI-generated content may be incorrect.