Mark Eilers

CS: 405 Secure Coding

4-1 Activity: Exceptions

March 30, 2024

4-1 Activity: Exceptions

The C++ code illustrates the implementation of exception handling mechanisms. Firstly, it defines a custom exception, CustomException, inheriting from std::exception, with a custom error message through the overridden what() method. Subsequently, two application logic functions are defined. do\_even\_more\_custom\_application\_logic() simulates a custom operation and throws a standard exception, while do\_custom\_application\_logic() encapsulates the former within a try-catch block to handle the exception and throws a CustomException. Additionally, a division function divide() is established, which throws an std::overflow\_error in case of a division by zero scenario. The handling functions do\_division() and main() demonstrate exception management. do\_division() attempts division and catches any potential overflow errors, printing the respective error message. In main(), both do\_division() and do\_custom\_application\_logic() are invoked within a try block, and exceptions are caught and handled accordingly. Specifically, CustomException and std::exception types are caught separately, with their corresponding error messages displayed, and a catch-all block is included to handle any other unforeseen exceptions. Overall, the code provides a comprehensive example of exception handling strategies in C++, encompassing custom exception creation, exception throwing, and robust error handling practices.

A computer screen with white text

Description automatically generated