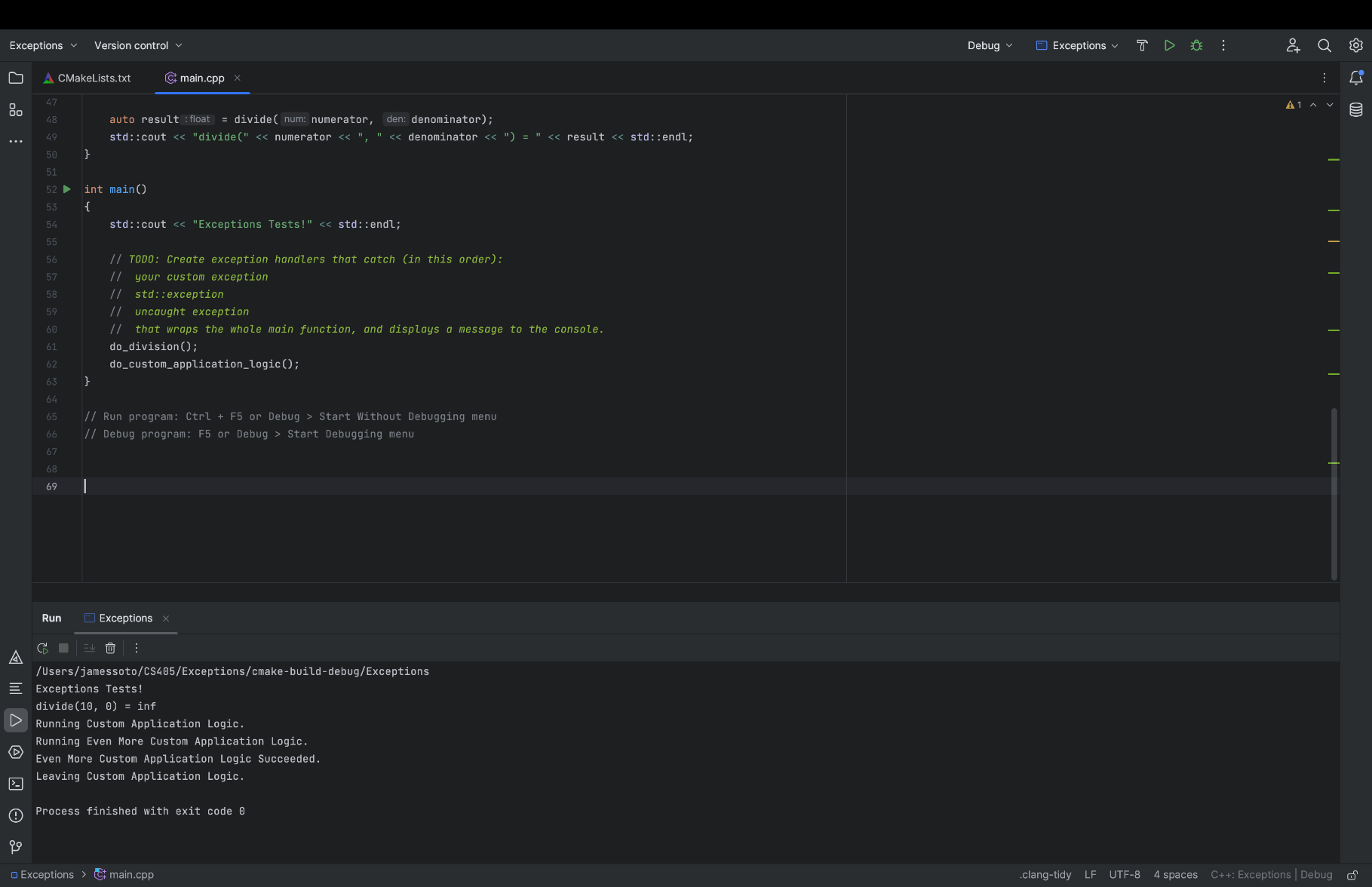
James Soto

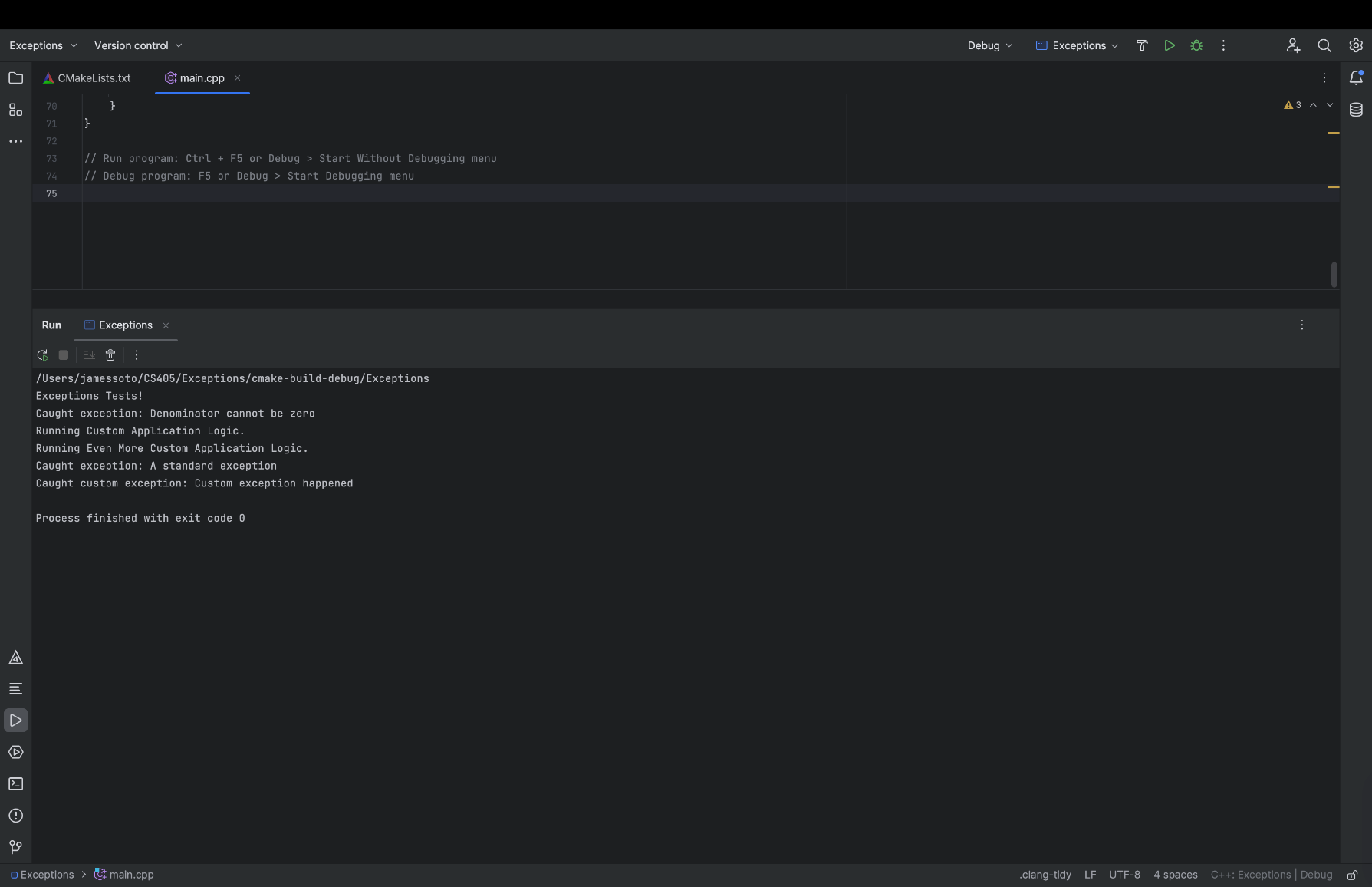
7-10-2023

CS-405 Secure Coding

4-1 Activity: Exceptions

Given code with output

Code output after changes



The code initially was not prepared to handle any exceptions, like the rubric guidelines mentions. To help this I introduced various try/catch blocks throughout the code to handle the required exceptions to meet the requirements from TODO. The function ‘do\_even\_more\_custom\_application\_logic()` was meant to throw a standard exception, but there was no such statement in place so I created a statement that would throw a standard runtime error that would simulate the error conditions. The function do\_custom\_application\_logic()` was meant to handle standard exceptions thrown by `do\_even\_more\_custom\_application\_logic()`, but there was no mechanism in place for this, so I introduced a try/catch block to catch and handle the standard exceptions thrown by `do\_even\_more\_custom\_application\_logic()`. In ‘do\_custom\_application\_logic()`, it was a requirement to throw a custom exception so I introduced the CustomException class that was derived from std::exception. The function ‘divide()’ was attempting to divide by zero, so I used an if statement to check that if the denominator is zero an exception is thrown using std::invalid\_argument. The ‘do\_division()’ function was calling the ‘divide()’ function without having any exception handling in place, so I introduced an try/catch block to handle exceptions thrown by ‘divide()’. The `main()` function was calling `do\_division()` and `do\_custom\_application\_logic()` without any exception handling in place, so I added a try/catch block to catch and handle the custom exception and the standard exceptions.