

A screenshot of a Visual Studio Debug Console window. The window title is "Microsoft Visual Studio Debug Console". The text inside the console is as follows:

```
Exceptions Tests!  
Division exception caught: Error: Division by zero  
Running Custom Application Logic.  
Running Even More Custom Application Logic.  
Standard exception caught: Application Logic Error  
Leaving Custom Application Logic.  
Runtime exception caught in main(): Custom Application Logic Exception Occurred.  
  
C:\Users\bryce\OneDrive - SNHU\CS-405\CS-405_M4_Exceptions\x64\Debug\CS-405_M4_Exceptions.exe (process 27980) exited with code 0.  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .
```

A new CustomException class, derived from std::exception, was created. It overrides the what() method to provide a custom error message. The custom exception is thrown from do_custom_application_logic() and caught explicitly in main().

The do_even_more_custom_application_logic() function now throws a std::runtime_error with message "Application Logic Error".

A try-catch block was added around the call to do_even_more_custom_application_logic() to catch any std::exception. It then outputs a message and the what() message from the exception. After this, a custom CustomException is thrown.

In the divide() function, a check for division by zero was added, throwing a std::invalid_argument exception when den == 0.

In the do_division() function a try-catch block was added in do_division() to catch

`std::invalid_argument` exceptions specifically, and an error message is displayed if the division by zero is attempted.

Using a try- catch block, exception handling was added in `main()` for: custom exceptions (specifically `CustomException`), standard exceptions (`std::exception`), any other uncaught exceptions (`catch(...)`).