

CS2530 - Lab Exception Properties – Part2

1. In Method1 add a try statement with a try and catch block
2. Inside the try block do the following:
 - Print the message "In Method1: calling Method2"
 - then call Method2
3. Catch an exception of type Exception and re-throw it after wrapping it in a new exception like this:

```
throw new Exception("Rethrowing exception in Method1 ", ex);
```

Before you re-throw the exception print the following message:

```
"In Method1 re-throwing the exception"
```
4. In Method3 inside the try block print the following statement before you call Convert.ToInt32

```
"In Method3 trying to convert a string to an integer"
```
5. Before you re-throw the exception in the catch statement write the following message:

```
"In Method3 re-throwing the exception"
```
6. In main replace the statements from part1 of the lab with the message and stack trace of
 - a) the exception and
 - b) the inner exception
 - c) the inner exception of the inner exception

Make sure to label each output appropriately.

Separate the message and the stack trace with a single empty line

Separate the three sections a) b) and c) with 2 empty lines for clarity.
7. Look at the line numbers in the stack traces and find the corresponding lines in the code.
8. **Write down the line numbers in the stack-traces of a) b) and c)** (exception, inner exception, and inner exception of the inner exception).
9. In the main method add a statement at the beginning of the catch block that prints the ToString method of the exception. Separate it with 2 empty lines from the following output.
10. Compare the line numbers in the stack-trace of the toString method with the line numbers in the stack-traces of a) b) and c)
11. **Write in a few words what you notice.**
12. In your program set three breakpoints where the three statements are that throw an exception. (Those are the two throw statements in the catch clauses and the original statement that triggered an

exception: `Convert.ToInt32)` .

You set a breakpoint by clicking on the light-blue bar left of the line numbers

13. Select Start Debugging (F5).

The execution will pause at the break point set next to `Convert.ToInt32`.

Ensure that the Call Stack window is open. If not you can open it like this:

Debug > Windows > Call Stack

Compare the line numbers in the Call Stack window to the line number you wrote down from the Console output.

14. Clicking F5 again will take you to the next breakpoint.

Once again compare the line numbers in the Call Stack window with the numbers your wrote down from the Console output.

15. Click F5 one more time. Repeat the line number comparison.

16. Notice how the information provided in the Call Stack Window help you find the method calls, that lead to the exception