Exception Demo

Demonstrate how to handle exceptions

# Training Objectives

* 1. Demonstrate how to write a try/catch/finally block.
  2. Demonstrate how to write a multiple catch block.

# Create a project

* 1. Create a new C# console application. Call it ExceptionDemo.

# Basic Try catch block

* 1. Add the following code to Main:

static void Main(string[] args)

{

string doAnother;

do

{

try

{

Console.Write("Please enter num1: ");

string num1str = Console.ReadLine();

int num1 = int.Parse(num1str);

Console.Write("Please enter num2: ");

string num2str = Console.ReadLine();

int num2 = int.Parse(num2str);

Console.WriteLine("Num1/Num2: {0}", num1 / num2);

}

catch

{

Console.WriteLine("There was an error!");

}

Console.Write("Do another (y/n): ");

doAnother = Console.ReadLine();

} while (doAnother == "y");

}

* 1. Try program with Ctrl-F5.
  2. Enter the following test cases:

|  |  |
| --- | --- |
| Num1 | Num2 |
| 5 | 2 |
| One | 5 |
| <blank> | 3 |
| 5 | 0 |
| 2147483647 | 5 |
| 2147483648 | 5 |

* 1. Your exceptions are caught but is this the best user experience? The user never knows what went wrong!

# Display error message and type

* 1. Change the catch block to:

catch (Exception exc)

{

Console.WriteLine("Error! "+exc.Message+"\nException type: "+exc.GetType());

}

* 1. Try the test cases out again. Notice that you can now see the types of exceptions that are thrown.
  2. Make a note of the various exception types.

# Add a finally block

* 1. After the catch block add:

finally

{

Console.WriteLine("This section always excecutes!");

}

* 1. Rerun the test cases. Notice that the finally block always executes, even after an exception!

# Catch Multiple Exceptions

* 1. The default exception messages do not always provide the best user experience. Catching individual exception types allows us to customize how information is provided to the user and how errors are handled.
  2. Add two more catch blocks before the generic Exception exc block:

catch(System.FormatException exc)

{

Console.WriteLine("Not a number!");

}

catch(System.OverflowException exc)

{

Console.WriteLine("Number is too large or small!");

}

* 1. Try the test cases again. You should have no unhandled exceptions. If you find one add another catch block.