Try/Catch Structures

Errors in C# are known as Exceptions.

Some kinds of exceptions can be anticipated, such as the user typing in the wrong data type, or leaving a field blank. But in more complex programs, there may be errors we can not anticipate, such as a conflict with other software, corrupted files or other unusual circumstances.

C# has a structure, **Try/Catch/Finally**, that is meant for catching these unanticipated errors, so that your program does not crash.

Syntax:

**try**

**{ }**

**catch**

**{ }**

**finally**

**{ }**

Try means "Try to execute this code".

Catch means "Catch any errors here".

Finally means "Do this code no matter what".

But you don’t always need to use all 3. A combination of try/catch/finally, try/catch, or try/finally will all work.

C# will TRY to execute the code in the first branch. If all goes well, it will skip the catch branch. But if there is an error, it will execute the code in the catch branch. Code in the finally branch will always be executed.

In your program, it would look something like this:

**try**

**{**

**//do some code here that might cause an error**

**}**

**catch**

**{**

**Console.WriteLine(“There was an error.”);**

**}**

**finally**

**{**

**Console.WriteLine(“This code runs no matter what.”);**

**}**

As you get into more sophisticated programming, you may begin to use more advanced features of Try/Catch/Finally in order to catch specific types of exceptions.

**BUT REMEMBER!!!!! We are practicing Try/Catch/Finally with common errors, only for convenience. You should normally deal with these using an IF structure. Only use Try/Catch/Finally for situations you can’t anticipate or code for, like file errors.**

More Details

You can also output a more detailed error message, using **(Exception e)** as parameters in the catch block.“e” can be any variableName, but you don’t declare it separately.

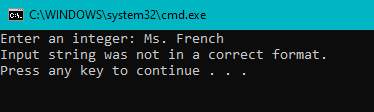
Syntax:

**catch(Exception e)**

**{**

**Console.WriteLine(e.Message);**

**}**

****

Activity

Write a Console program, **TryCatch**, that allows you to enter two integers and then tries to divide them. Use the code shown on the next page!

Download a copy of this document.

Test the inputs listed below in your Try/Catch program, then paste your results right in this document:

* Two reasonable integers, ex. 10 and 5

Result: 2

* A super long integer, ex. 999999999999999999999999

Result: Value was either too large or too small for an Int32

* Any normal integer, and a 0

Result: Attempted to divide by zero.

* A string

Result: Input string was not in a correct format.

**//attempt to get 2 integers and divide them.**

**try**

**{**

**Console.Write("Enter a number: ");**

**int num1 = int.Parse(Console.ReadLine());**

**Console.Write("Enter another number: ");**

**int num2 = int.Parse(Console.ReadLine());**

**int result = num1 / num2;**

**Console.WriteLine("Result: " + result);**

**}**

**//this block runs if there is an error**

**catch (Exception ex)**

**{**

**Console.WriteLine(ex.Message);**

**}**