

Lesson 3-5: Understand the Exception object

So far all of your code may have run without any problems, but unfortunately that won't always be so. When things do go wrong, Visual Studio will do its best to tell you what the problem is.

You've already seen a build error in *Lesson 1-8: Run a project*, but those only appear when Visual Studio is able to detect a problem before the code runs.

Exceptions appear when something goes wrong while the code is running. By looking at the exception, you will hopefully be able to work out what the problem is.

- 1 Open *CSharpTest* from your sample files folder.
- 2 Open *debugme.aspx.cs* (The code-behind file of *debugme.aspx*).

There is some code in here that you won't recognize, but the important thing is that it's going to cause an error by trying to divide by zero.

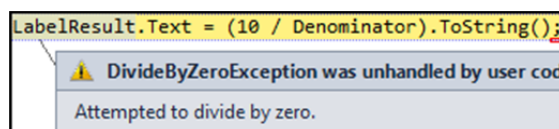
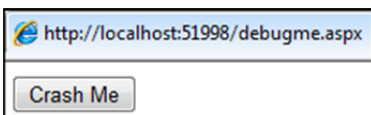
You will understand everything that this code is doing after: *Lesson 5-10: Perform basic mathematical operations*.

- 3 Cause an exception.

1. Start debugging to view *debugme.aspx* in Debug mode.
2. Click the *Crash Me* button.

You should be brought back to Visual Studio. If not, switch back to it manually without closing the browser window.

You will see the line that caused the exception highlighted in yellow and a box showing the details of the exception.



This exception is very obvious. The error message at the top tells you that it was caused by attempting to divide by zero.

Unfortunately not all error messages are as easy to understand.

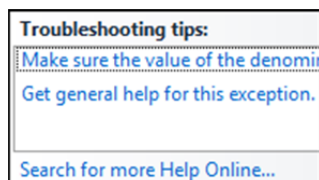
- 4 Use troubleshooting tips.

You'll notice that the *Exception* window shows a list of *Troubleshooting tips*. You can click on them to read articles relevant to the problem, which might help you to fix it.

You can also click *Search for more Help Online...* to be redirected to a search page which will allow you to search Microsoft's articles for anything relevant.

- 5 View the details of the exception.

An exception isn't just an error message; it has properties, just like the controls on your page.



Actions:

View Detail...

Enable editing

Copy exception detail to the clipboard

note**null**

You can see two *null* values for properties in the screenshots of the *Exception* object and you might have noticed them in a few other properties you saw earlier.

null is a value that literally means 'nothing'. A property with a value of *null* is considered not to have a value.

It's important to note that there's a difference between 0 and *null*, since 0 is a number and *null* is the absence of a number.

You'll learn more about *null* in: *Lesson 5-12: Understand null.*

1. Click the *View Detail...* link in the *Exception* box.
2. Expand *System.DivideByZeroException* by clicking the arrow.

System.DivideByZeroException	{"Attempted to divide by zero."}
[System.DivideByZeroException]	{"Attempted to divide by zero."}
▶ Data	{System.Collections.ListDictionar
HelpLink	null
▶ InnerException	null
Message	Attempted to divide by zero.
Source	CSharpTest
StackTrace	at CSharpTest.debugme.Button
▶ TargetSite	{Void ButtonCrash_Click(System

The most important properties of the exception are *InnerException*, *Message*, *Source* and *StackTrace*.

InnerException will contain another exception if this one had a deeper cause. Essentially, the Inner Exception is the exception that caused this one. Since this exception didn't have a deeper cause, the Inner Exception is *null* (see sidebar).

Message is the error message you've already seen displayed on the main exception box.

Source is the *namespace* that caused the error. You'll learn about namespaces in: *Lesson 6-4: Work with namespaces.*

StackTrace shows the operation that caused the error. If you read it you'll see that the first line tells you that it originated in *debugme.ButtonCrash_Click*.

6 Fix the problem.

This error happened because the number the code divided by was zero. You can fix this by changing the number.

1. Click *OK* and stop debugging.

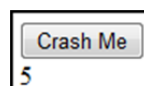
Either close your web browser or click *Debug*→*Stop Debugging*.

2. Change the line *int Denominator = 0;* to:

int Denominator = 2;

```
protected void ButtonCrash_Click(object sender, EventArgs e)
{
    int Denominator = 2;
    LabelResult.Text = (10 / Denominator).ToString();
}
```

3. View the page in *Debug* mode and click the button.



You will see that there is no longer an error. Instead the result of 10 divided by 2 is displayed.