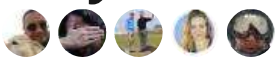
ow to use finally blocks

/2017 • 2 minutes to read •  +4

s article

lso

n an exception occurs, execution stops and control is given to the appropriate tion handler. This often means that lines of code you expect to be executed are ssed. Some resource cleanup, such as closing a file, needs to be done even if an tion is thrown. To do this, you can use a `finally` block. A `finally` block always ites, regardless of whether an exception is thrown.

ollowing code example uses a `try/catch` block to catch an mentOutOfRangeException. The `Main` method creates two arrays and attempts to one to the other. The action generates an ArgumentOutOfRangeException and the is written to the console. The `finally` block executes regardless of the outcome of opy action.

Copy

```
.ng System;

ass ArgumentOutOfRangeExample

  public static void Main()
  {
      int[] array1 = {0, 0};
      int[] array2 = {0, 0};

      try
      {
          Array.Copy(array1, array2, -1);
      }
      catch (ArgumentOutOfRangeException e)
      {
          Console.WriteLine("Error: {0}", e);
          throw;
      }
      finally
      {
          Console.WriteLine("This statement is always executed.");
      }
  }
```

e also

Exceptions

his page helpful?

Yes No

commended content

sing keyword - C# Reference

sing keyword - C# Reference

ceptions and Exception Handling - C# Programming Guide

earn about exceptions and exception handling. These C# features help deal with nexpected or exceptional situations that happen when a program is running.

y-finally - C# Reference

y-finally - C# Reference

f keyword - C# Reference

f keyword - C# Reference

Show more