

## Contents

<b>Exceptions (Solutions)</b>	<b>1</b>
Questions . . . . .	1
Warm-up Exercise . . . . .	2

## Exceptions (Solutions)

### Questions

1. In computing, what is an exception?
  - A compilation error.
  - When a program user requires a special accommodation.
  - When a behaviour not supposed to happen occurs during execution.
  - A keyword.
2. When a program meets an unexpected behaviour, we say that it...
  - ... *raises* an exception.
  - ... *throws* an exception.
  - All of the above.
3. An exception can occur when...
  - ... a user enters for example the **string** "Test" when asked for a numerical value.
  - ... a division by 0 occurs.
  - ... the program tries to access an array outside of its index range.
  - All of the above.
4. A **try-catch** block...
  - ... executes all the code inside its **try** block, then all its code inside its **catch** block.
  - ... executes all the code inside its **try** block, then all its code inside its **catch** block if an exception was raised at any point.
  - ... executes only if an exception was raised in the program before.
  - ... executes the code inside its **try** block, and switches to its **catch** block if an exception was thrown.
  - ... executes its **catch** block first, and then its **try** block if an exception was raised.
5. A **try-catch-finally** block...
  - ... can have multiple **catch** block.

- ... can omit the **finally** block.
- ... can omit the **catch** block.
- All of the above.

## Warm-up Exercise

1. Consider the following code:

```

using System;

class Program
{
    static void Main()
    {
        try
        {
            Console.WriteLine("Enter a number");
            int uInput =
                ↪ int.Parse(Console.ReadLine());
            int[] t = { 10 };
            int div = 0 / (uInput -1);
            int tAcces = t[uInput];
        }
        catch (IndexOutOfRangeException)
        {
            ↪ Console.WriteLine("IndexOutOfRangeException");
        }
        catch (DivideByZeroException)
        {
            ↪ Console.WriteLine("DivideByZeroException");
        }
        catch (FormatException)
        {
            Console.WriteLine("FormatException");
        }
        catch (ArgumentNullException)
        {
            ↪ Console.WriteLine("ArgumentNullException");
        }
    }
}

```

(Download this code)<sup>1</sup>

- Determine which input would the user needs to enter to get "IndexOutOfRangeException", "DivideByZeroException", "FormatException" and "ArgumentNullException" displayed.
- Is there something the user could enter that would *not* raise any exception?

Solution

Exception	Input
"IndexOutOfRangeException"	Any number greater than 2.
"DivideByZeroException"	1
"FormatException"	Any string that is not a number (for example, "Test")
"ArgumentNullException"	A <b>null</b> string (ctrl + d on linux, ctrl + z on windows)

Entering 0 would not raise any exception.

---

<sup>1</sup><https://princomp.github.io/code/projects/TriggeringExceptions.zip>