

Day12 Assignment

By

Narala Praveen

08-Feb-2022

Question1:

What is Exception Handling and why we need exception handling?

Exception Handling: Exception handling is a process to handle the run time errors occurred during the working of an application.

Needs for exception handling:

Exception handling is done to ensure that the application not to crash (or)

Will not display any technical details to make sure we handle errors gracefully and display friendly messages.

Question2:

Write a simple division program and handle three exceptions discussed in the class ,also add super exception at the last?

Program:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

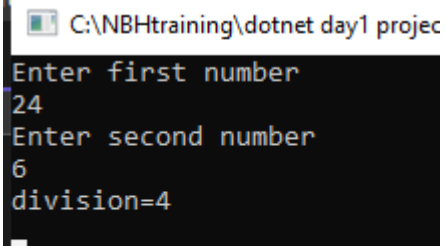
namespace exceptionproject1
{
    //Author:Narala Praveen
    //Purpose:To write a simple division program with three exceptions
    internal class Program
    {
        static void Main(string[] args)
        {
            //we can write number of methods in "Try"
            try
            {
                int a;
                int b;
                int c;

                Console.WriteLine("Enter first number");
                a=Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Enter second number");
                b=Convert.ToInt32(Console.ReadLine());
                c = a / b;
                Console.WriteLine("division={0}", c);
                Console.ReadLine();
            }
            //Exceptions:
            catch (OverflowException)
            {
                Console.WriteLine("The number entered out of range");
                Console.ReadLine();
            }
            catch (FormatException)
            {
                Console.WriteLine("The entered input is in wrong format enter
```

```

only numbers");
    Console.ReadLine();
}
catch(DivideByZeroException)
{
    Console.WriteLine("The number can't be divide by zero");
    Console.ReadLine();
}
catch(Exception)
{
    Console.WriteLine("some error occured please contact
@Praveencompany");
    Console.ReadLine();
}
}
}
}

```



C:\NBHtraining\dotnet day1 projec

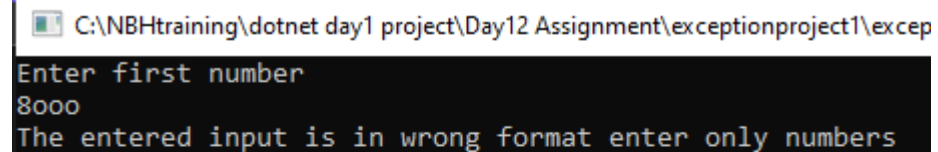
```

Enter first number
24
Enter second number
6
division=4
_

```

Correct output:

Format Exception:



C:\NBHtraining\dotnet day1 project\Day12 Assignment\exceptionproject1\excep

```

Enter first number
8000
The entered input is in wrong format enter only numbers

```

Divide by Zero Exception:

```
C:\NBHtraining\dotnet day1 project\Day12 Assignn
Enter first number
5
Enter second number
0
The number can't be divide by zero
```

[illegible]

Overflow exception:

Super Exception:

```
C:\NBHtraining\dotnet day1 project\Day12 Assignment\exceptionproject
Enter first number
6
Enter second number
0
some error occured please contact @Praveencompany
```

Question3:

Research and write at least 6 exceptions that occur in c# with sample code?

Invalid Cast type castrating:

```
namespace Exceptionsproject
{
    class Praveen
    {
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Praveen a = new Praveen();
            object b = a;
            Program c=(Program)b;
        }
    }
}
```

Exception Unhandled

System.InvalidCastException: 'Unable to cast object of type 'Exceptionsproject.Praveen' to type 'Exceptionsproject.Program'.'

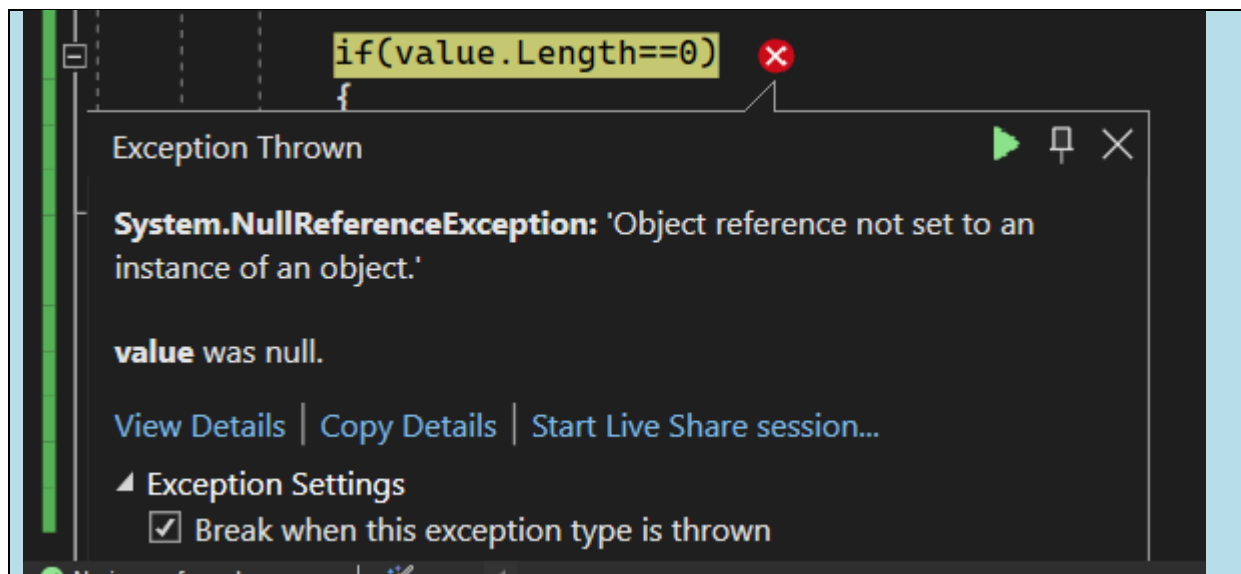
[View Details](#) | [Copy Details](#) | [Start Live Share session...](#)

▶ [Exception Settings](#)

Null reference exception:

```
internal class Program
{
    static void Main(string[] args)
    {
        string value=null;

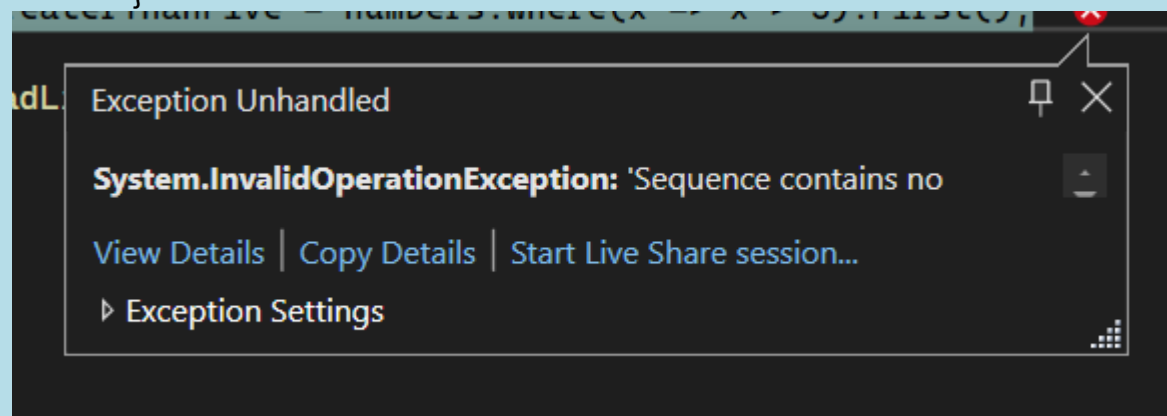
        if(value.Length==0)
        {
            Console.WriteLine(value);
        }
        Console.ReadLine();
    }
}
```



Invalid Operations Exception:

```
static void Main(string[] args)
{
    var numbers = new List<int> { 2, 4, 5 };
    var firstgreaterThanFive = numbers.Where(x => x > 5).First();

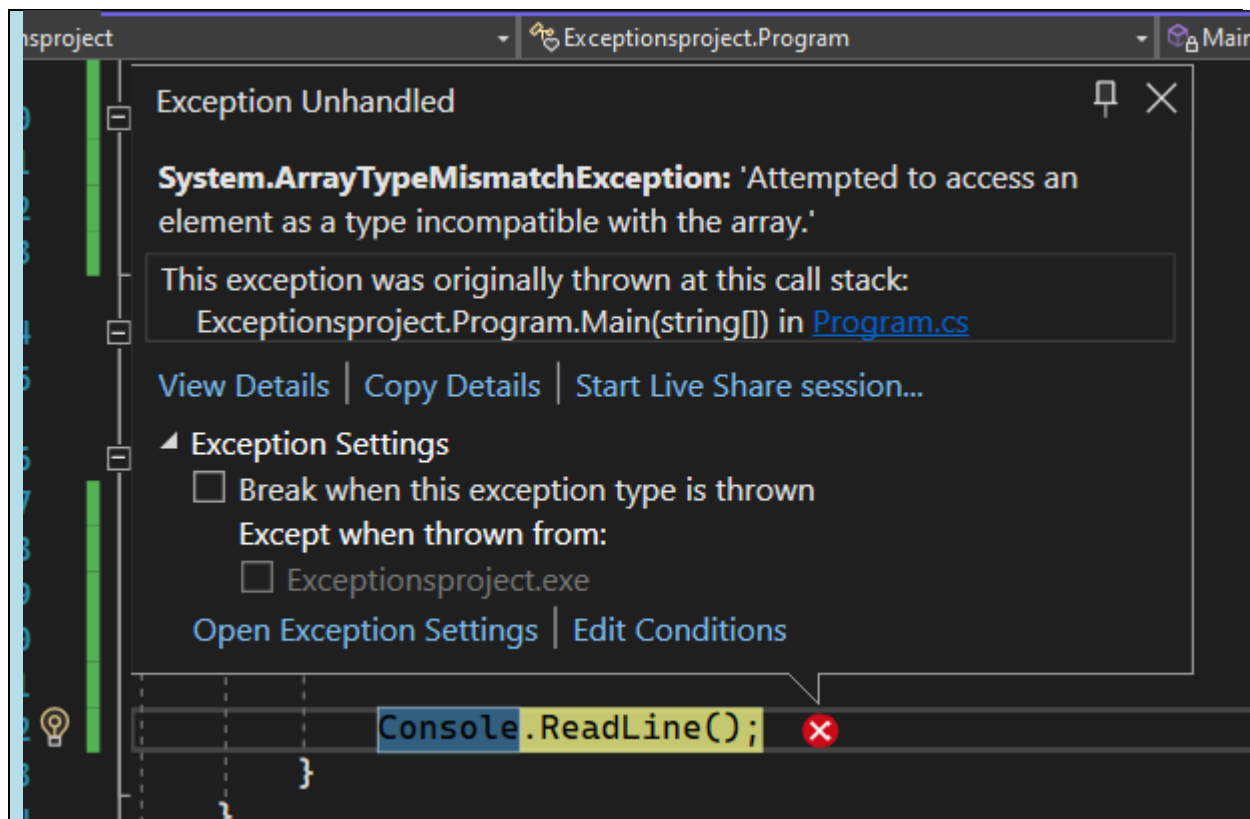
    Console.ReadLine();
}
```



Array type mismatch Exception:

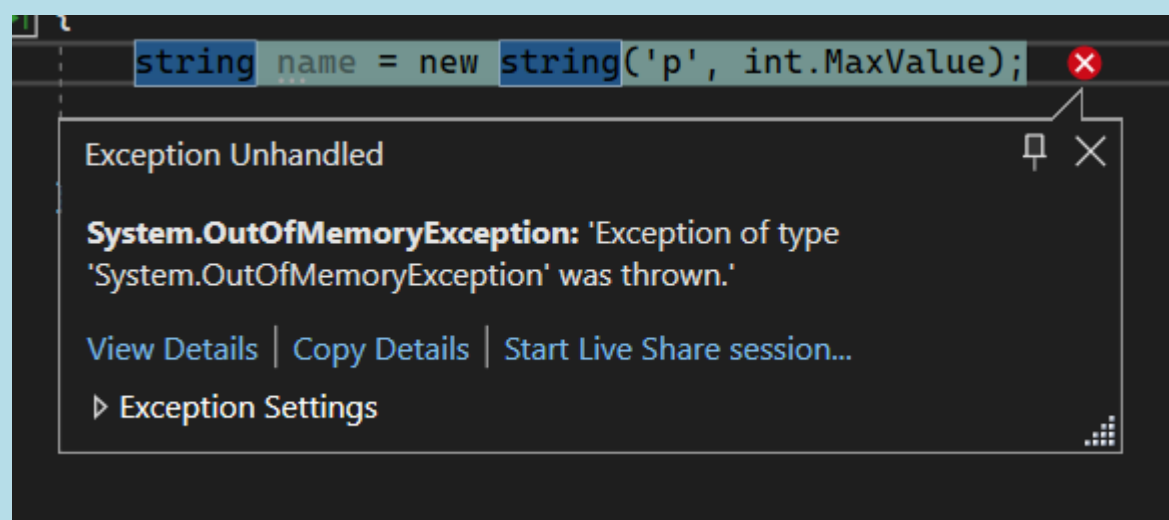
```
static void Main(string[] args)
{
    string[] a = { "hi" };
    object[] b = a;
    b[0] = 3;

    Console.ReadLine();
}
```



Out of memory Exceptions: `internal class Program`

```
{  
    static void Main(string[] args)  
    {  
        string name = new string('p', int.MaxValue);  
  
        Console.ReadLine();  
    }  
}
```



Question4:

What is the use of "finally" block illustrate with an example?

Statements in final block will be executed irrespective of errors occurred or not will be executed.

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace finallyproject3
//Author:Narala Praveen
//Purpose:example for finally block
{
    internal class Program
    {
        static void Main(string[] args)
        {
            //we can write number of methods in "Try"
            try
            {
                int a;
                int b;
                int c;

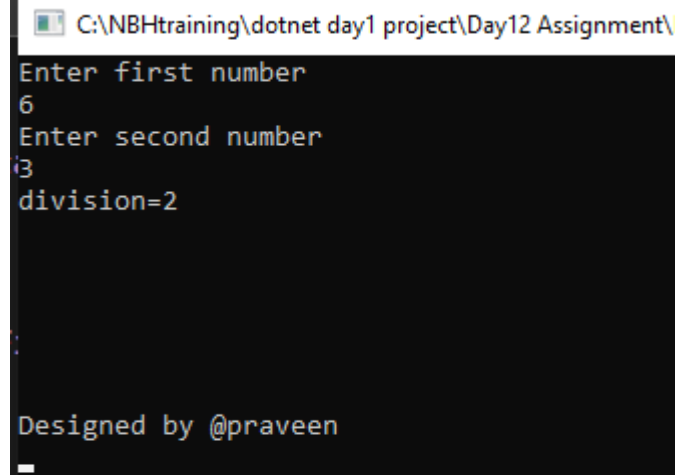
                Console.WriteLine("Enter first number");
                a = Convert.ToInt32(Console.ReadLine());
                Console.WriteLine("Enter second number");
                b = Convert.ToInt32(Console.ReadLine());
                c = a / b;
                Console.WriteLine("division={0}", c);

            }
            //Exceptions:
            catch (OverflowException)
            {
                Console.WriteLine("The number entered out of range");
            }
            catch (FormatException)
            {
                Console.WriteLine("The entered input is in wrong format enter
only numbers");
            }
            catch (DivideByZeroException)
            {
                Console.WriteLine("The number can't be divide by zero");
            }
            catch (Exception)
            {
                Console.WriteLine("some error occurred please contact
@Praveencompany");
            }
            finally
            {
                Console.WriteLine("\n\n\n\n\n\nDesigned by @praveen");
                Console.ReadLine();
            }
        }
    }
}
```



```
}  
  
    }  
}  
}
```

Output:



```
C:\NBHtraining\dotnet day1 project\Day12 Assignment\  
Enter first number  
6  
Enter second number  
3  
division=2  
  
Designed by @praveen  
_
```

Question5:**Write 5 points about exception handling discussed in class?**

- a. Exception handling is done to ensure that application not crash.**
- b. A single try block can have multiple catches .**
- c. General exception should be at last.**
- d. Statements in final block will be executed irrespective of errors occurred or not will be executed.**

```
static void Main(string[] args)
{
    try
    {
    }
    catch
    {
    }
    finally
    {
    }
}
```

- e.** Console.ReadLine();

Question6:

What is compilation error and Runtime error, write at least 3 differences

Compilation error: When we compile the code, the errors we get when the code does not get compiled are called compilation errors.

Runtime Error: The errors that occur during the execution of the program are called run time errors.

Difference between runtime and compilation error:

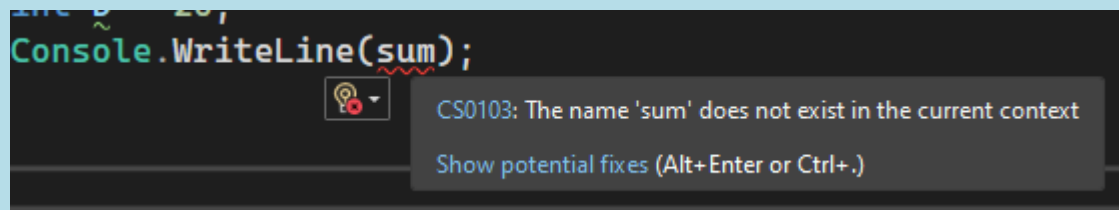
Runtime Error	Compile Error
It occurs during execution	It occurs during compiling of code
These are difficult to identify	This are easy to identify
These are logic primarily.	These are basically syntax errors.

Question7:

Write any six compilation errors with small code snippet.
Add compilation error screen shots?

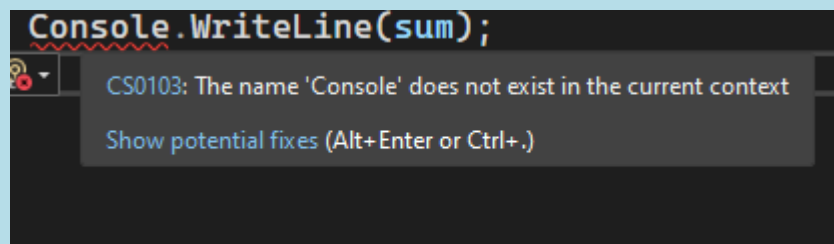
Compile error1:

```
static void Main(string[] args)
{
    int a = 10;
    int b = 20;
    Console.WriteLine(sum);
}
```



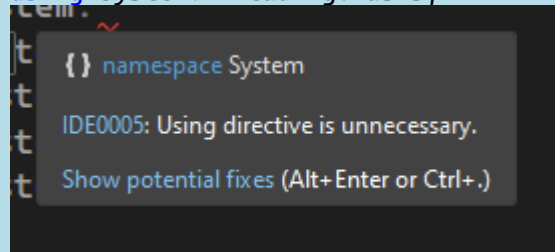
Compile error2:

```
static void Main(string[] args)
{
    int a = 10;
    int b = 20;
    int sum = 0;
    sum = a + b;
    Console.WriteLine(sum);
}
```



Compile error3:


```
using System.
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```



Compile error4:

```
int a = 10;
string b = 20;
int sum = 0;

Console.WriteLine(sum);
```

 `struct System.Int32`

Represents a 32-bit signed integer. To browse the .NET Framework source code for this type, see the Reference Source.

CS0029: Cannot implicitly convert type 'int' to 'string'

Compile error5:

```
static void Main(string[] args)
{
    int input;
    public void Readdata()
    {
        Console.WriteLine("Enter input");
    }
}
```

`public void Readdata()`

CS0106: The modifier 'public' is not valid for this item

Compile error6:

```
static void Main(string[] args)
{
    List<int> list = new List<int>();
    pot.Add(90);
    pot.Add(90);
    pot.Add(90);

    Console.WriteLine();
}
```

`pot.Add(90);`



CS0103: The name 'pot' does not exist in the current context

Show potential fixes (Alt+Enter or Ctrl+.)

`Console.WriteLine();`

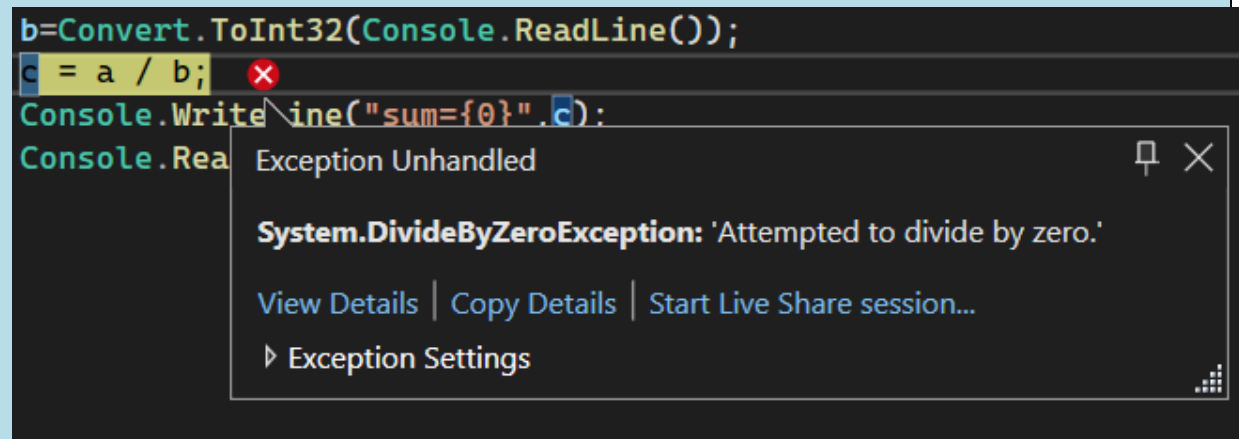
Question8:

Write any 6 runtime errors with small code snippets and add run time error screen shots?

Runtime error 1:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Exceptionsproject
{
    class Praveen
    {
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            int a;
            int b;
            int c;
            Console.WriteLine("Enter first number");
            a=Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter second number");
            b=Convert.ToInt32(Console.ReadLine());
            c = a / b;
            Console.WriteLine("sum={0}",c);
            Console.ReadLine();
        }
    }
}
```



Runtime error2:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

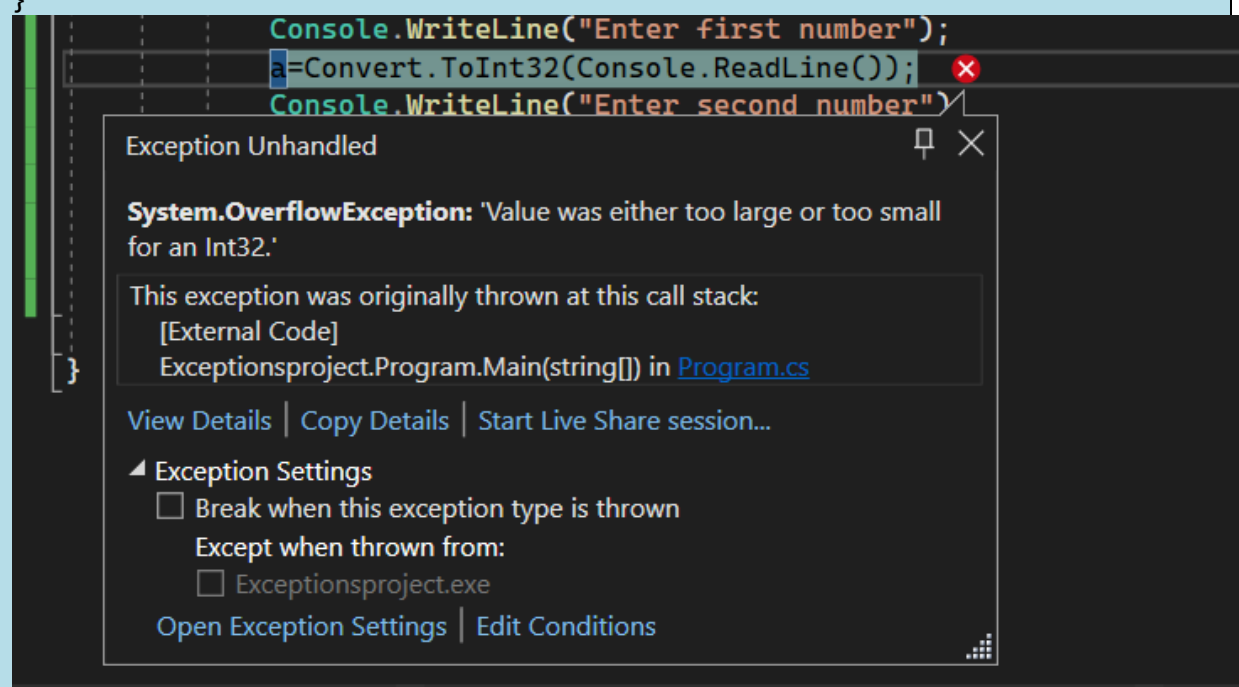
namespace Exceptionsproject
```

```

{

    internal class Program
    {
        static void Main(string[] args)
        {
            int a;
            int b;
            int c;
            Console.WriteLine("Enter first number");
            a=Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter second number");
            b=Convert.ToInt32(Console.ReadLine());
            c = a / b;
            Console.WriteLine("sum={0}",c);
            Console.ReadLine();
        }
    }
}

```



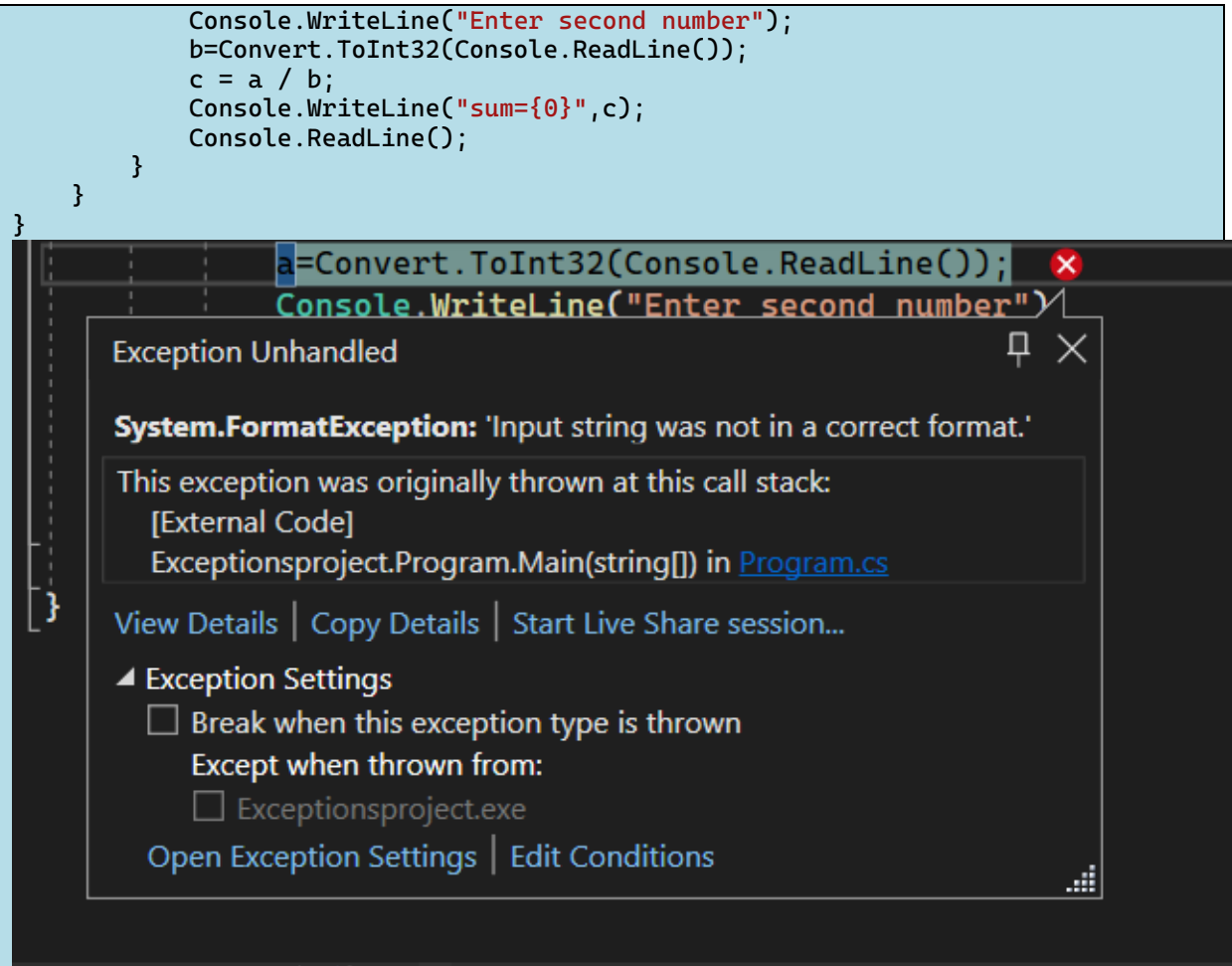
Runtime error3:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Exceptionsproject
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int a;
            int b;
            int c;
            Console.WriteLine("Enter first number");
            a=Convert.ToInt32(Console.ReadLine());

```



Runtime error4:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Exceptionsproject
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[] Array = new int[6];
            Array[8] = 20;
            Console.WriteLine(Array[8]);
            Console.ReadLine();
        }
    }
}
```



```
ay = new int[6];
```

```
= 20;
```

```
WriteLine(Array[8]):
```

```
ReadLine
```

Exception Unhandled

System.IndexOutOfRangeException: 'Index was outside the bounds of the array.'

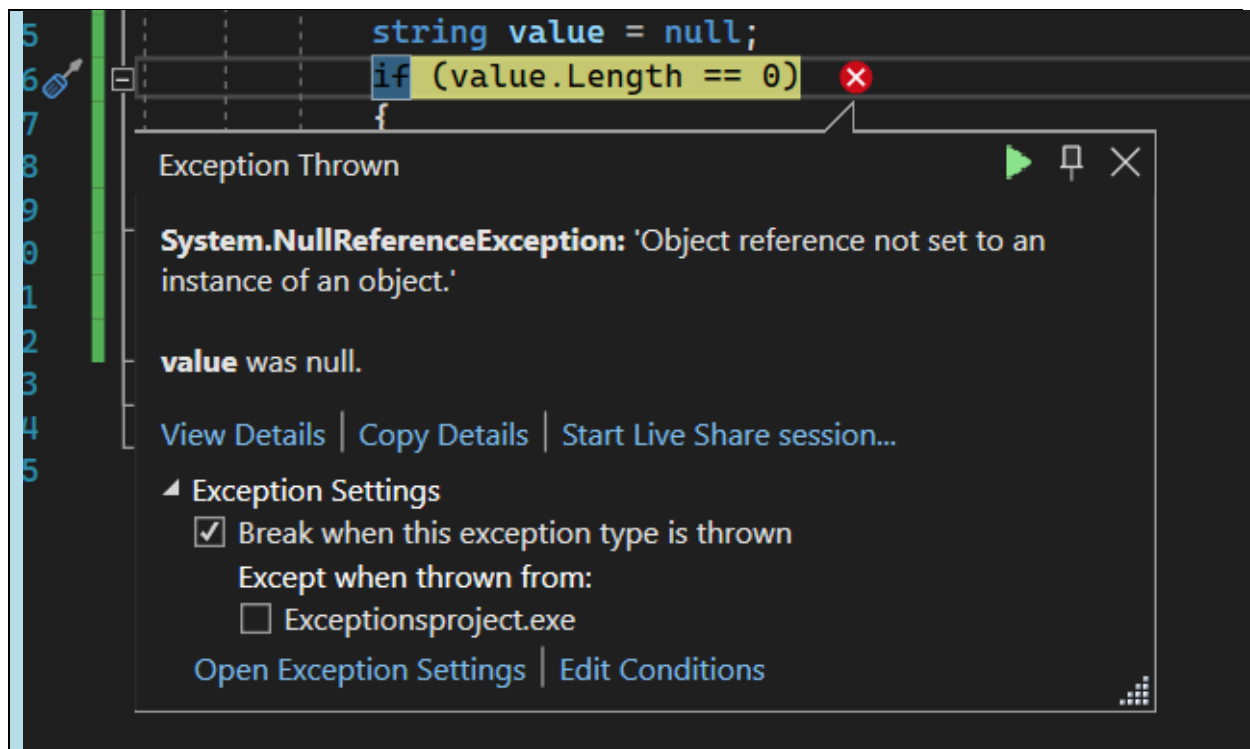
[View Details](#) | [Copy Details](#) | [Start Live Share session...](#)

▶ [Exception Settings](#)

Runtime error5:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Exceptionsproject
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string value = null;
            if (value.Length == 0)
            {
                Console.WriteLine(value);
            }
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```



Runtime error6:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Exceptionsproject
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string[] a = { "Shankar" };
            object[] b = a;
            b[0] = 3;
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```

