Day 12 Assignment
By
Vinay Kudali
08-02-22

1. What is Exception Handling and why we need exception handling.

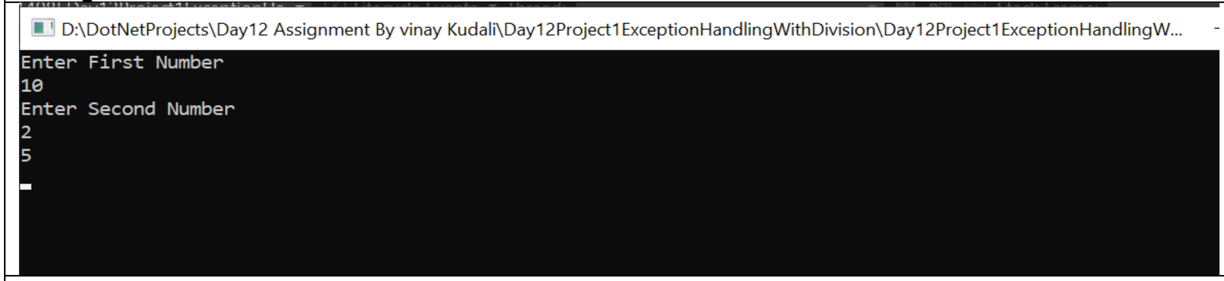
An exception is a problem that arises during the execution of a program. Exception handling is done to ensure that our application will not crash or will not display any technical details and to make sure we handle errors gracefully and display friendly messages.

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

Code:

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

namespace Day12Project1ExceptionHandlingWithDivision
{
   internal class Program
   {
      //Author Vinay Kudali
      //Purpose: Creating 3 Type of exceptions with division
      static void Main(string[] args)
      {
         try
         {
            int a, b, 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(c);
            Console.ReadLine();
         }
         catch (DivideByZeroException ex) //this is the catch block for divide by zero exception
         {
            Console.WriteLine("Denominator Shouldn't be Zero : Enter Number Which Is not Zero");
         }
         catch (OverflowException ex) //this is the catch block for over flow exception exception
         {
            Console.WriteLine("Enter The value with in the range : 1 to 50000");
         }
         catch (FormatException ex) // this is the catch block For format Exception
         {
            Console.WriteLine("input should be numeric value");

         }
         catch (Exception ex)// this is Super Exception
         {
            Console.WriteLine("Some error occured");
         }
         Console.ReadLine();
      }
   }
}
```

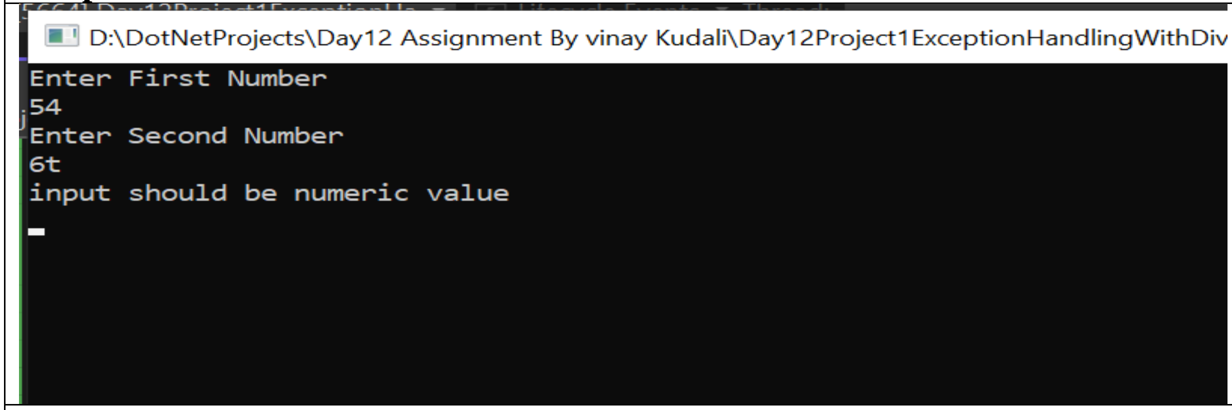
Output 1:



D:\DotNetProjects\Day12 Assignment By vinay Kudali\Day12Project1ExceptionHandlingWithDivision\Day12Project1ExceptionHandlingW...

```
Enter First Number
10
Enter Second Number
2
5
```

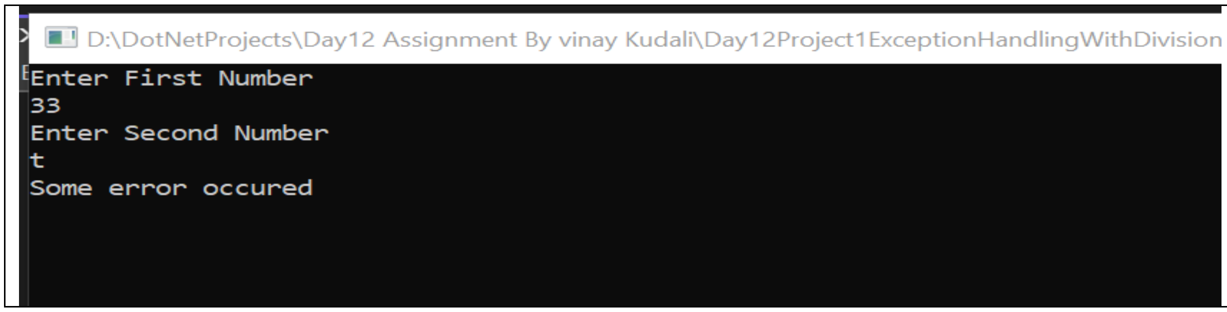
Output2:



D:\DotNetProjects\Day12 Assignment By vinay Kudali\Day12Project1ExceptionHandlingWithDiv

```
Enter First Number
54
Enter Second Number
6t
input should be numeric value
```

Output3:



D:\DotNetProjects\Day12 Assignment By vinay Kudali\Day12Project1ExceptionHandlingWithDivision

```
Enter First Number
33
Enter Second Number
t
Some error occured
```

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

“**Finally**” is a keyword. The code inside “**finally Block**” will get executed regardless whether or not there is an Exception.

Uses: By using a **finally block,** we can clean up any resources that are allocated in a try block, and we can run code even if an exception occurs in the **try block.**

5. Write the 5 points I explained about exception handling.

1. Exception Handling is done to handle Exceptions or errors Gracefully without displaying any errors to the end customers.
2. A single try block can have multiple catch block
3. A general Exception Should be written at the last.
4. Statements inside the finally block will be executed irrespective of Whether exception Occurs or not.
5. A general Syntax Of exception handling is **try, catch, finally.**

6. What is compilation and Runtime error? Write at least 3 differences between them.

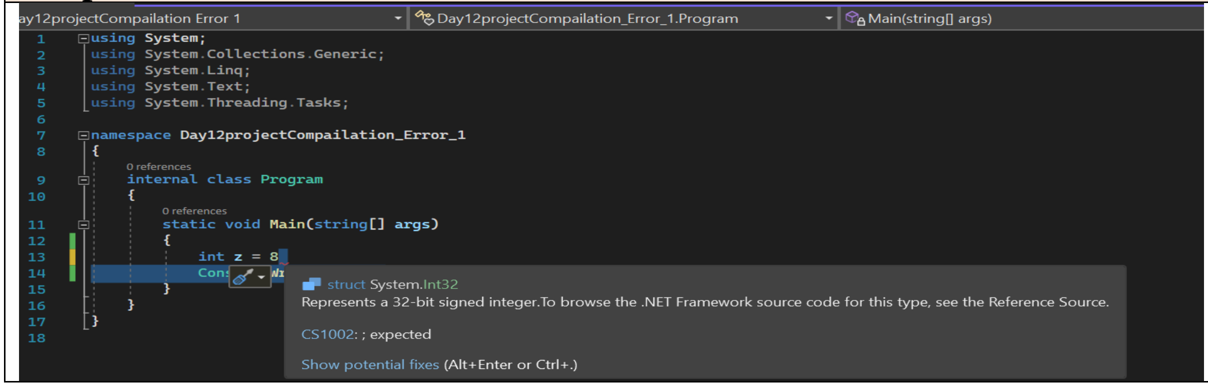
Compilation Errors	Run time Errors
• To find the compilation errors, Developer no need to be involve. It will show Red Line.	• To find the Run time errors, Developer needs to be involved. There is no need for indication to find the run time errors.
• It has a physical property; we can see compilation error before compilation.	• We can find it after compilation done by the developer. It will run in back ground execution.
• These are the syntax errors which are detected by the compiler.	• These are errors which are not detected by the compiler and produce wrong results.

7. Write any 6 compilation errors with a small code snippet. Add compilation error screen shots.

1)

```
internal class Program
    {
        static void Main(string[] args)
        {
            int z = 8
            Console.WriteLine(z);
        }
    }
```

Output:



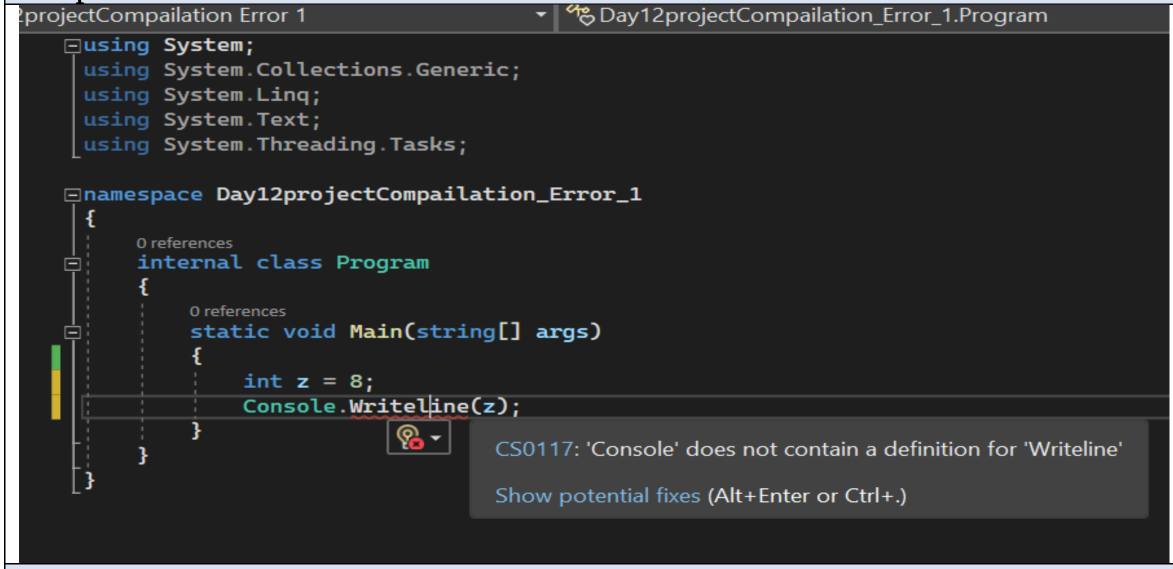
```
ay12projectCompailation Error 1    Day12projectCompailation_Error_1.Program    Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day12projectCompailation_Error_1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int z = 8
            Cons
        }
    }
}
```

```
struct System.Int32
Represents a 32-bit signed integer.To browse the .NET Framework source code for this type, see the Reference Source.

CS1002: ; expected

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

2)

```
internal class Program
    {
        static void Main(string[] args)
        {
            int z = 8;
            Console.Writeline(z);
        }
    }
```

Output:



```
?projectCompailation Error 1                Day12projectCompailation_Error_1.Program
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day12projectCompailation_Error_1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int z = 8;
            Console.Writeline(z);
        }
    }
}
```

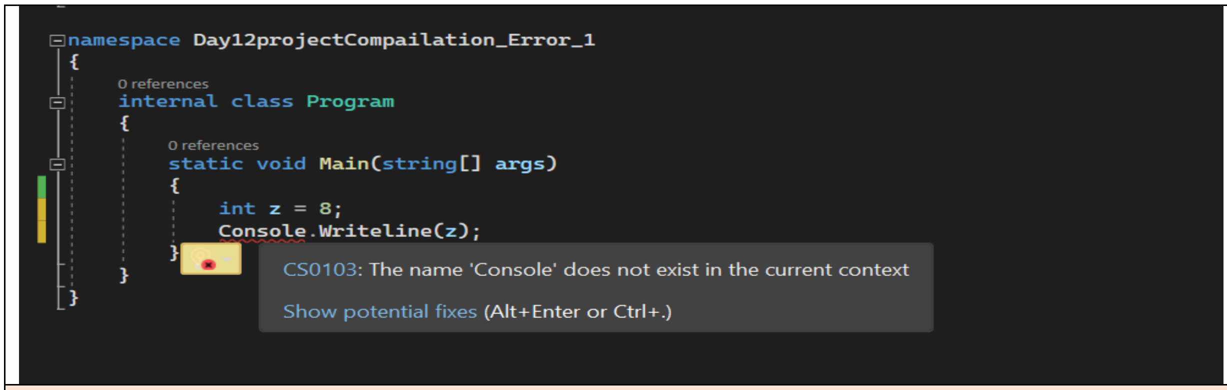
CS0117: 'Console' does not contain a definition for 'Writeline'

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

3)

```
internal class Program
 {
     static void Main(string[] args)
     {
         int z = 8;
         Console.Writeline(z);
     }
 }
```

Output:



```
namespace Day12projectCompailation_Error_1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int z = 8;
            Console.Writeline(z);
        }
    }
}
```

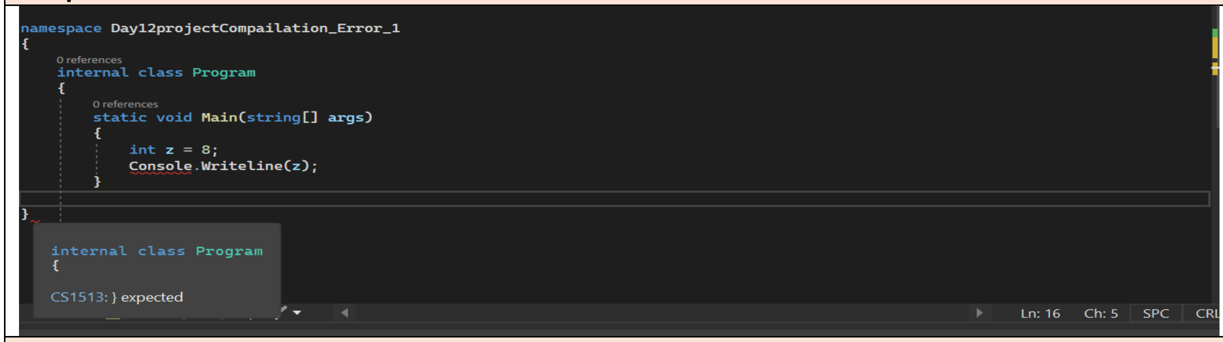
CS0103: The name 'Console' does not exist in the current context

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

4)

```
internal class Program
   {
      static void Main(string[] args)
      {
         int z = 8;
         Console.Writeline(z);
      }

}
```

Output:



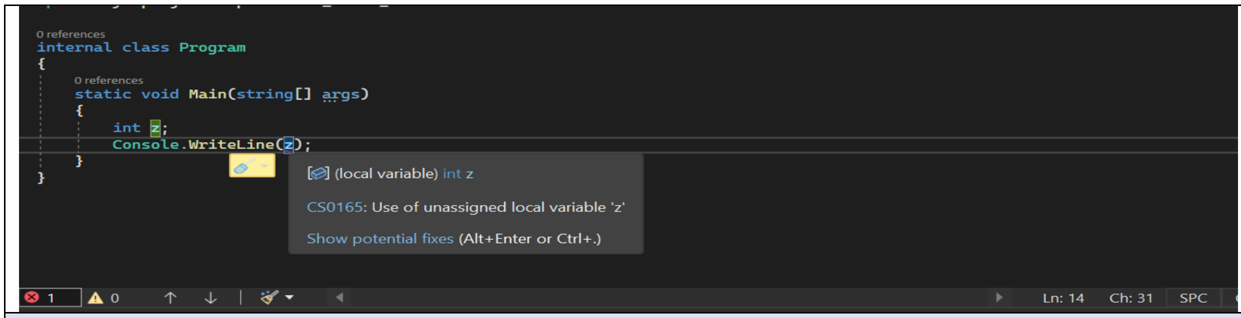
```
namespace Day12projectCompailation_Error_1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int z = 8;
            Console.Writeline(z);
        }

}
```

```
internal class Program
{

CS1513: } expected
```

5)

```
internal class Program
   {
      static void Main(string[] args)
      {
         int z;
         Console.WriteLine(z);
      }
   }
```

Output:



```
0 references
internal class Program
{
    0 references
    static void Main(string[] args)
    {
        int z;
        Console.WriteLine(z);
    }
}
```

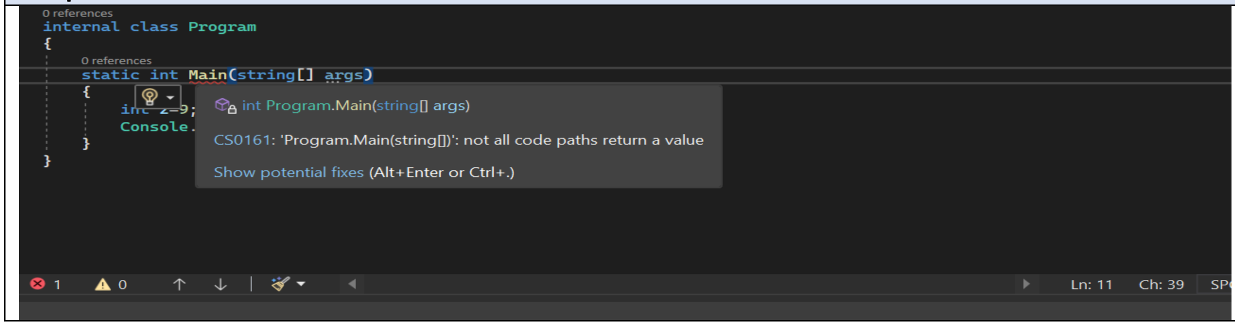
```
(local variable) int z
CS0165: Use of unassigned local variable 'z'
Show potential fixes (Alt+Enter or Ctrl+.)
```

```
1   0                                  Ln: 14   Ch: 31   SPC
```

6)

```
{
       static int Main(string[] args)
       {
          int z=9;
          Console.WriteLine(z);
       }
   }
```

Output:



```
0 references
internal class Program
{
    0 references
    static int Main(string[] args)
    {
        int z=9;
        Console.
    }
}
```

```
int Program.Main(string[] args)
CS0161: 'Program.Main(string[])': not all code paths return a value
Show potential fixes (Alt+Enter or Ctrl+.)
```

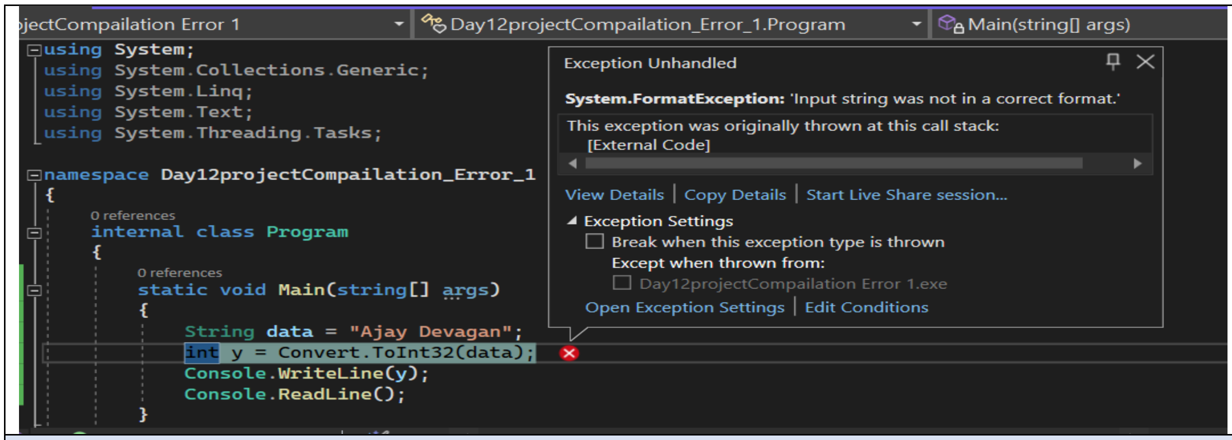
```
1   0                                  Ln: 11   Ch: 39   SP
```

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

1) System.FormatException

```
internal class Program
   {
       static void Main(string[] args)
       {
           String data = "Ajay Devagan";
           int y = Convert.ToInt32(data);
           Console.WriteLine(y);
           Console.ReadLine();
       }
```

Output:

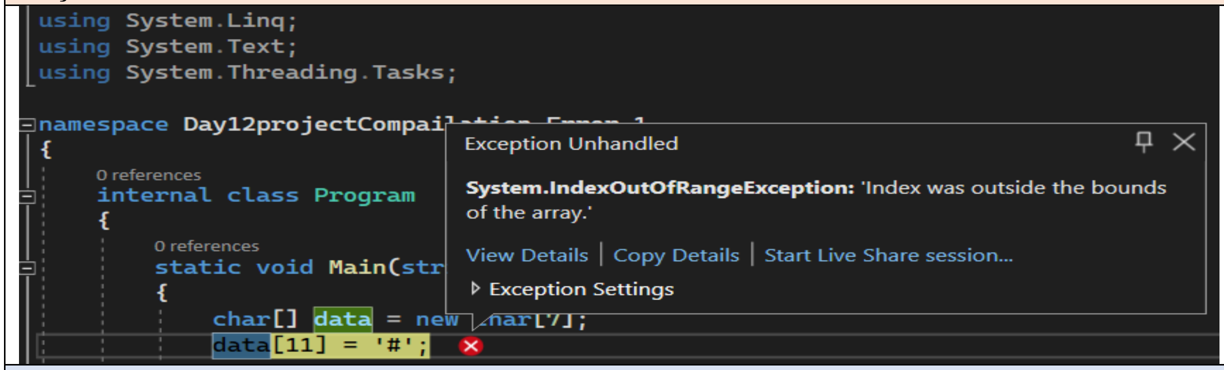


```
jectCompailation Error 1    Day12projectCompailation_Error_1.Program    Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day12projectCompailation_Error_1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            String data = "Ajay Devagan";
            int y = Convert.ToInt32(data);
            Console.WriteLine(y);
            Console.ReadLine();
        }
```

```
Exception Unhandled

System.FormatException: 'Input string was not in a correct format.'

This exception was originally thrown at this call stack:
    [External Code]

View Details | Copy Details | Start Live Share session...
Exception Settings
  ☐ Break when this exception type is thrown
    Except when thrown from:
      ☐ Day12projectCompailation Error 1.exe
  Open Exception Settings | Edit Conditions
```

2) System.OutOfRangeIndexException

```
internal class Program
   {
       static void Main(string[] args)
       {
           char[] data = new char[7];
           data[11] = '#';

       }
   }
```



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

namespace Day12projectCompailation_Error_1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(str
        {
            char[] data = new char[7];
            data[11] = '#';
```

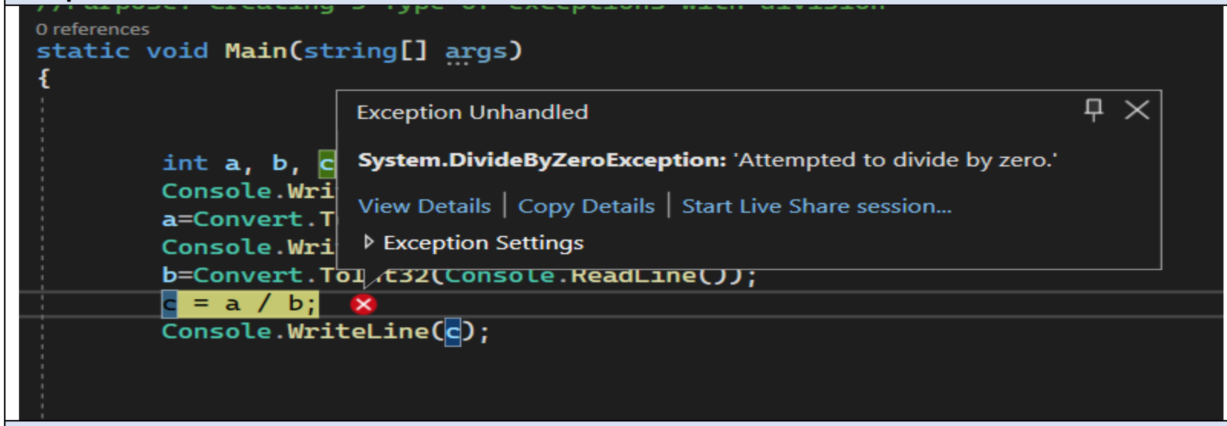
```
Exception Unhandled

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

View Details | Copy Details | Start Live Share session...
Exception Settings
```

3) System.DivideByZeroException

```
static void Main(string[] args)
     {



          int a, b, 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(c);
```

```
        Console.ReadLine();
    }
```

Output:



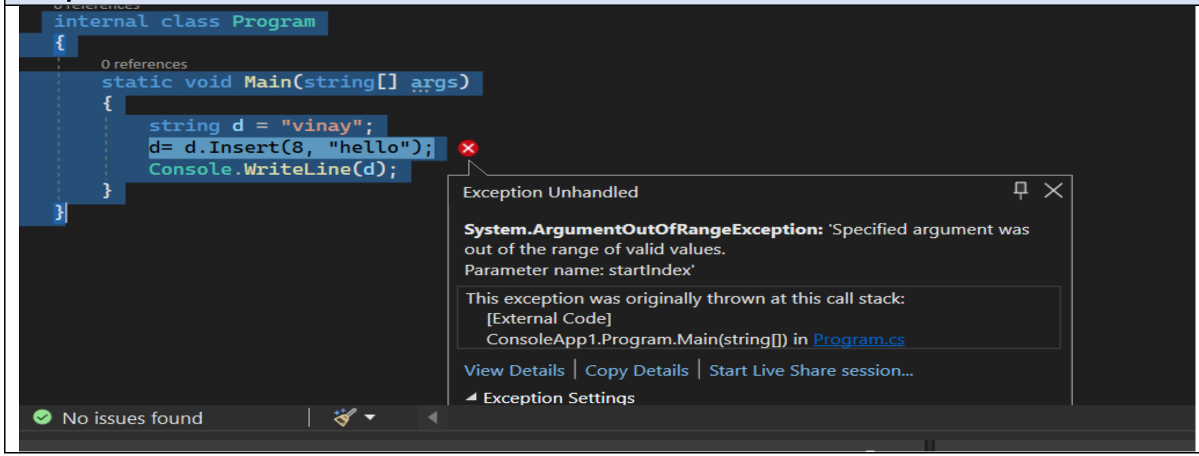
```
0 references
static void Main(string[] args)
{

        int a, b, c
        Console.Wri
        a=Convert.T
        Console.Wri
        b=Convert.ToI  t32(Console.ReadLine());
        c = a / b;
        Console.WriteLine(c);
```

```
Exception Unhandled

System.DivideByZeroException: 'Attempted to divide by zero.'

View Details | Copy Details | Start Live Share session...
▷ Exception Settings
```

4) System.ArgumentOutOfRangeException:

```
internal class Program
  {
     static void Main(string[] args)
     {
        string d = "vinay";
        d= d.Insert(8, "hello");
        Console.WriteLine(d);
     }
  }
```

Output:



```
internal class Program
{
    0 references
    static void Main(string[] args)
    {
        string d = "vinay";
        d= d.Insert(8, "hello");
        Console.WriteLine(d);
    }
}
```

```
Exception Unhandled

System.ArgumentOutOfRangeException: 'Specified argument was
out of the range of valid values.
Parameter name: startIndex'

This exception was originally thrown at this call stack:
    [External Code]
    ConsoleApp1.Program.Main(string[]) in Program.cs

View Details | Copy Details | Start Live Share session...
◢ Exception Settings
```