Feb **8th** Morning Assignment

By Surya Teja Chandolu

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

Exception Handling:

An exception is an occurrence that occurs during the execution of a programme that the programme code does not expect. The measures to be taken in the event of an exception are unknown to the programme.

As part of C#, you can use the try, catch, and finally keywords to try actions that may not succeed, to deal with failures when you determine it is necessary to do.

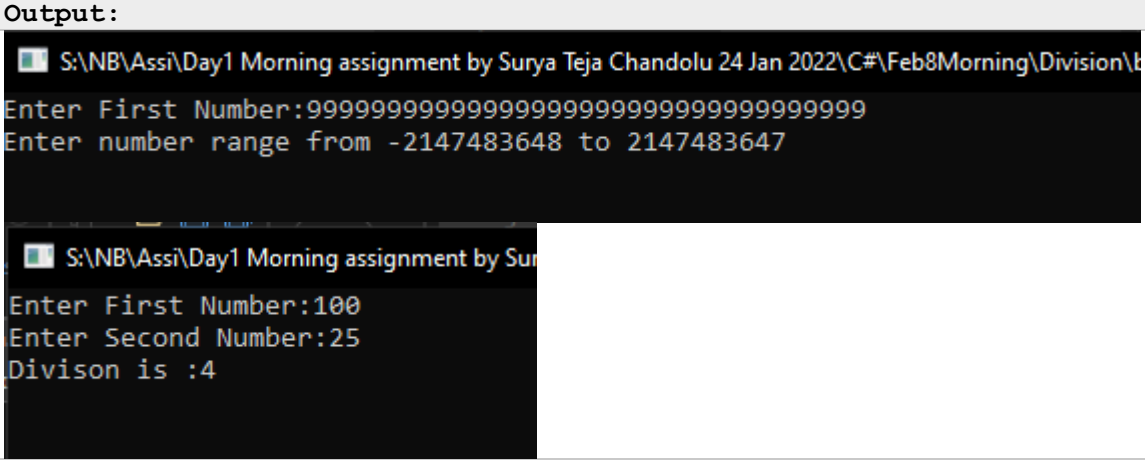
2. Write a simple division program and handle three exceptions discussed in the class., also add super exception at the last.

Code:

```
using System;

/*********************************************************************
* Author: Surya Teja
* Purpose: Divison of two numbers using try, catch and finally block
* *******************************************************************/

namespace Division
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int firstNumber, secondNumber, div;

            try
            {
                Console.Write("Enter First Number:");
                firstNumber = Convert.ToInt32(Console.ReadLine());

                Console.Write("Enter Second Number:");
                secondNumber = Convert.ToInt32(Console.ReadLine());

                div = firstNumber / secondNumber;

                Console.WriteLine($"Divison is :{div}");
            }
            catch (OverflowException)
            {
                Console.WriteLine($"Enter number range from {int.MinValue} to
{int.MaxValue}");
            }
            catch (DivideByZeroException)
            {
                Console.WriteLine("Enter second number as other than zero");
            }
```

```
            catch (FormatException)
            {
                Console.WriteLine("Enter only numbers");
            }
            catch (Exception)
            {
                Console.WriteLine("Please contact Office");
            }
            finally
            {
                Console.ReadLine();
            }
        }
    }
}
```

Output:



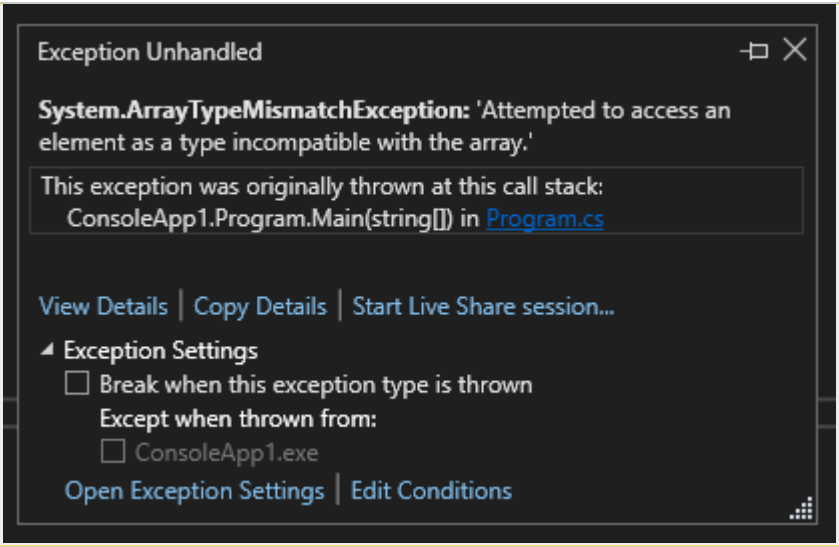
3. Research and write atleast six exceptions that occur in C# with sample code.

ArrayTypeMismatchException

Code:

```
using System;

namespace ConsoleApp1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string[] name = { "Surya" };
            object[] names = name;
            names[0] = 3;

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

Exception:

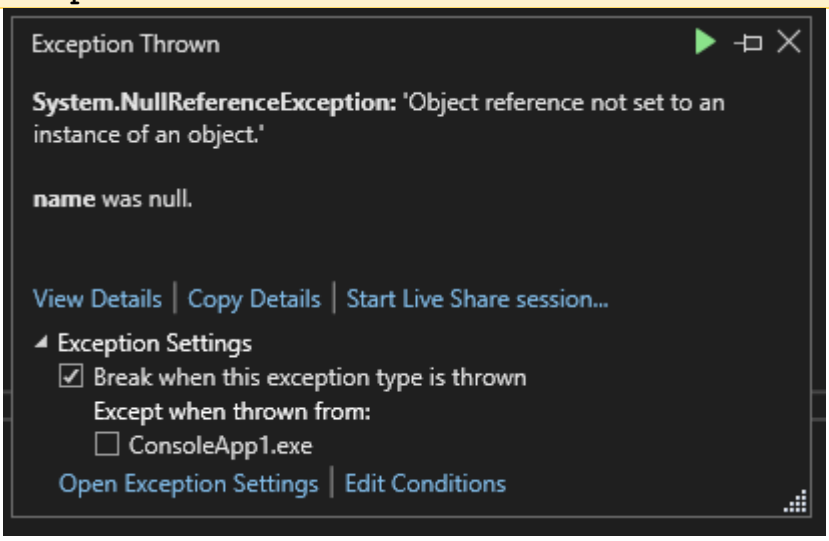


NullReferenceException

Code:

```
using System;

namespace ConsoleApp1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string name = null;

            Console.WriteLine(name);

            if(name.Length == 0)
                Console.WriteLine(name);

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

Exception:

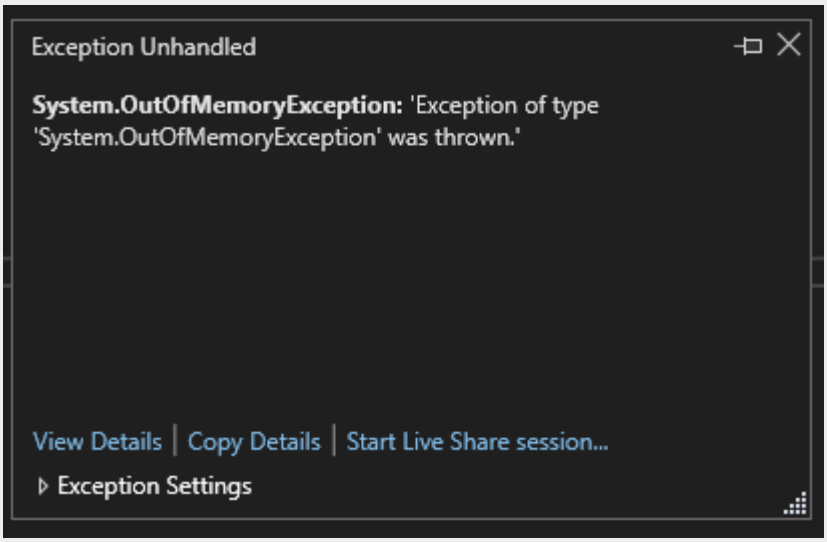


OutOfMemoryException

Code:

```
using System;

namespace ConsoleApp1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            string name = new string('S', int.MaxValue);

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

Exception:

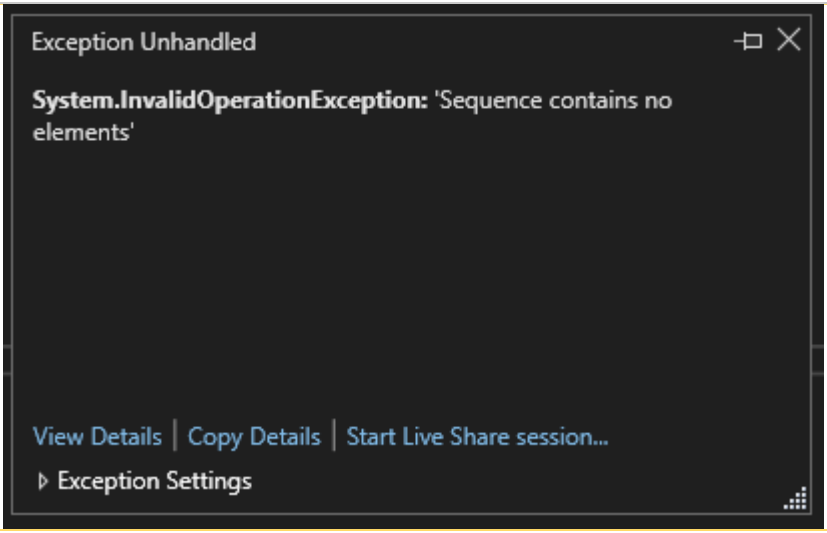


InvalidOperationException

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace ConsoleApp1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            var num = new List<int> { 1, 5, 10 };
            var gt = num.Where(x => x > 10).First();

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

Exception

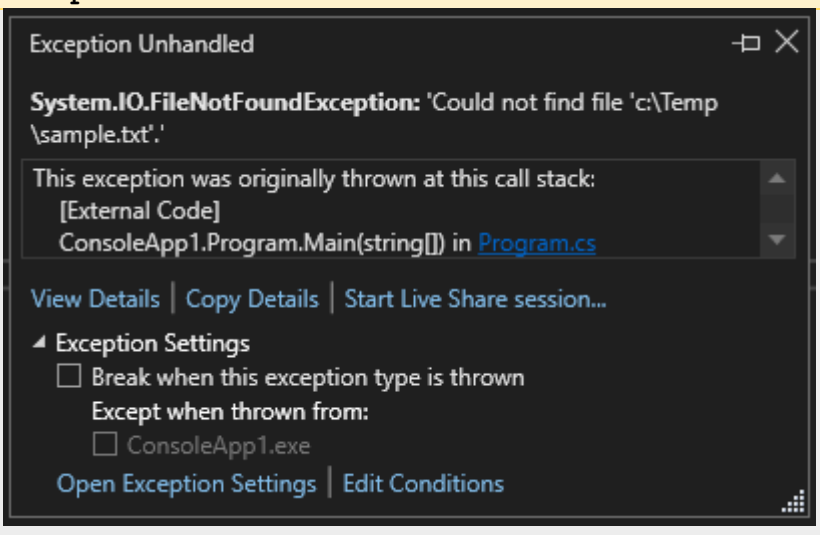


FileNotFoundException

Code:

```
using System;
using System.IO;

namespace ConsoleApp1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            StreamReader sr = new StreamReader(@"c:\Temp\sample.txt");
            Console.WriteLine(sr.ReadToEnd());

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

Exception:

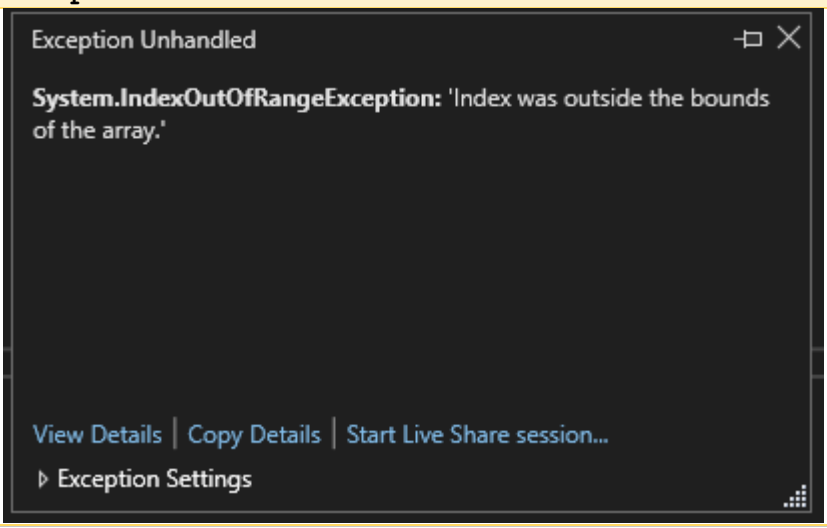


IndexOutOFRangeException

Code:

```
using System;

namespace ConsoleApp1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int [] data = new int[3];
            data[0] = 10;
            data[1] = 20;
            data[2] = 30;
            data[3] = 40;
            data[4] = 50;

            foreach (int i in data)
                Console.WriteLine($"{i} ");

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

Exception:



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

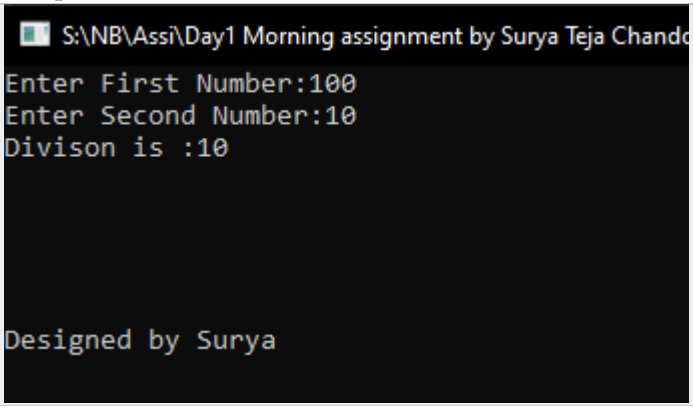
Finally Block: A finally block contains all statements that must be executed regardless of whether or not an exception arises. Regardless of whether an exception happens in the try block or not, the statements in this block.

Example:

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

/*************************************************************************
* Author: Surya Teja
* Purpose: Divison of two numbers using try, catch and finally block
* ************************************************************************/

namespace Division
```

```
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int firstNumber, secondNumber, div;

            try
            {
                Console.Write("Enter First Number:");
                firstNumber = Convert.ToInt32(Console.ReadLine());

                Console.Write("Enter Second Number:");
                secondNumber = Convert.ToInt32(Console.ReadLine());

                div = firstNumber / secondNumber;

                Console.WriteLine($"Divison is :{div}");
            }
            catch (OverflowException)
            {
                Console.WriteLine($"Enter number range from {int.MinValue} to
{int.MaxValue}");
            }
            catch (DivideByZeroException)
            {
                Console.WriteLine("Enter second number as other than zero");
            }
            catch (FormatException)
            {
                Console.WriteLine("Enter only numbers");
            }
            catch (Exception)
            {
                Console.WriteLine("Please contact Office");
            }
            finally
            {
                Console.WriteLine("\n\n\n\n\nDesigned by Surya");
                Console.ReadLine();
            }
        }
    }
}
```

Output:



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

- Exception handling is done to handle errors gracefully and not to crash.
- In Exception handling one try block can have multiple catch block.
- We must initialize Exception only at last.
- Finally block will execute all statements whether or not an exception arises.
- Try → Catch → Finally.

6. What is compilation and Runtime error Write atleast 3 differences between them

Compile Time Error	Run Time Error
Errors that occur when you break the rules of writing syntax.	Errors occur while executing the program.
Errors are detected by compiler which are easy to fix.	Errors are hard to find and fix the issue.
It prevent running code with errors.	It run the code and display the wrong output.
EX: missing {}, ;, “”, etc,.	EX: dividing by 0.

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

Input num is not initlize.

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

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int num;
            Console.WriteLine(num);
        }
    }
}
```

(local variable) int num

CS0165: Use of unassigned local variable 'num'

Class name is missing

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

namespace ConsoleApp1
{
    internal class
    {
        0 references
        static void Main(string[] args)
        {
            int num =
            Console.Wr
        }
    }
}
```

void <invalid-global-code>.Main(string[] args)

CS0116: A namespace cannot directly contain members such as fields, methods or statements

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

In static keyword S should be in lower case.

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

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        Static void Main(string[] args)
        {
        }
    }
}
```

IDE1007: The name 'Static' does not exist in the current context.

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

System class is not added.

```
namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int num = 10;
            Console.WriteLine(num);
        }
    }
}
```

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

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

Cannot implement string to int.

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

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int num;
            Console.WriteLine("Enter number");
            num = Console.ReadLine();

            Console.WriteLine(
        }
    }
}
```

```
string Console.ReadLine()
Reads the next line of characters from the standard input stream.

Returns:
The next line of characters from the input stream, or null if no more lines are available.

Exceptions:
System.IO.IOException
OutOfMemoryException
ArgumentOutOfRangeException

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

```
1   0
rom: Debug
```

Statement must end with ;

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

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int num;
            Console.WriteLine("Enter number");
            num = Convert.ToInt32(Console.ReadLine())

            Console.WriteLine(num);
        }
    }
}
```

```
CS1002: ; expected

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

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

Number cannot divide by zero

```
using System;

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int num1 = 20, num2 = 0, div;
            div = num1 / num2;

            Console.WriteLine(div);
        }
    }
}
```

Exception Unhandled

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

View Details | Copy Details | Start Live Share session...

Exception Settings

Name cannot convert into integer.

```
using System;

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            string name = "surya";
            int value = Convert.ToInt32(name);

            Console.WriteLine(value);

            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]
ConsoleApp1.Program.Main(string[]) in Program.cs

View Details | Copy Details | Start Live Share session...

Exception Settings
Break when this exception type is thrown
Except when thrown from:
ConsoleApp1.exe
Open Exception Settings | Edit Conditions

Integer number is long

```
using System;

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int num1, num2, sum;
            Console.WriteLine("Enter first number");
            num1 = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter second number");
            num2 = Convert.ToInt32(Console.ReadLine());
            sum = num1 + num2;
            Console.WriteLine(sum);

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

Exception Unhandled

System.OverflowException: 'Value was either too large or too small for an Int32.'

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
Break when this exception type is thrown
Except when thrown from:
ConsoleApp1.exe
Open Exception Settings | Edit Conditions

Array index is out of range

```
using System;

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            int [] data = new int[3];
            data[0] = 10;
            data[1] = 20;
            data[2] = 30;
            data[3] = 40;
            data[4] = 50;

            foreach (int i in data)
                Console.WriteLine($"{i} ");

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

```
Exception Unhandled

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

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

Length of Null value cannot be zero

```
using System;

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            string name = null;

            if(name.Length == 0)
                Console.WriteLine(name);

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

```
Exception Thrown

System.NullReferenceException: 'Object reference not set to an instance of an object.'

name was null.

View Details | Copy Details | Start Live Share session...
Exception Settings
Break when this exception type is thrown
Except when thrown from:
ConsoleApp1.exe
Open Exception Settings | Edit Conditions
```

Array type mismatch

```
using System;

namespace ConsoleApp1
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            string[] name = { "Surya" };
            object[] names = name;
            names[0] = 3;

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

```
Exception Unhandled

System.ArrayTypeMismatchException: 'Attempted to access an element as a type incompatible with the array.'

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

View Details | Copy Details | Start Live Share session...
Exception Settings
Break when this exception type is thrown
Except when thrown from:
ConsoleApp1.exe
Open Exception Settings | Edit Conditions
```