

Day 12 Assignment  
By  
M Mary Margarette  
On 08-02-2022

What is Exception Handling and why we need exception handling?

→ 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.

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

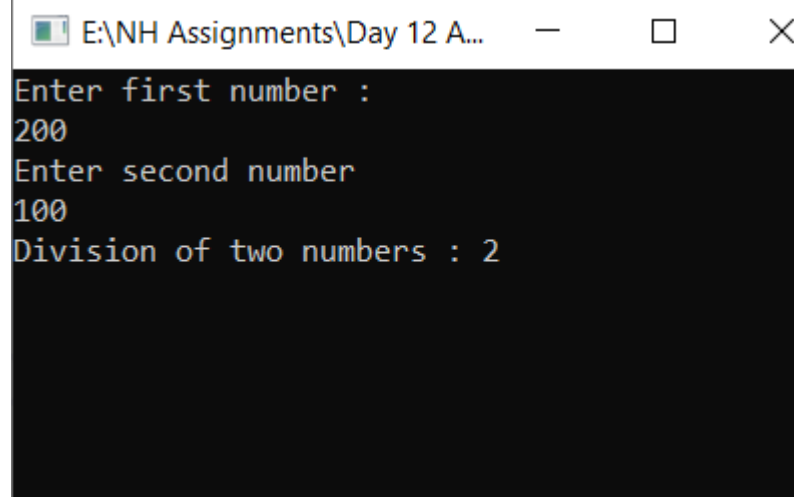
Code:

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

namespace Day_12_Project_1
{
    internal class Program
    {
        //Author : Mary Margaret
        //Exception Handling
        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("Division of two numbers : {0} ", c);
                Console.ReadLine();
            }
        }
    }
}
```

```
        catch (OverflowException)//exception for large numbers
        {
            Console.WriteLine("please give numbers between 0 and 700000");
            Console.ReadLine();
        }
        catch (DivideByZeroException)//exception when divide with zero
        {
            Console.WriteLine("Number Can't divided by zero");
            Console.ReadLine();
        }
        catch (FormatException)//exception when format is not correct
        {
            Console.WriteLine("sorry only numbers are accepted, please check again");
            Console.ReadLine();
        }
        catch (Exception)//super exception
        {
            Console.WriteLine("something went wrong please contact admin@nbh.com");
            Console.ReadLine();
        }
    }
}
```

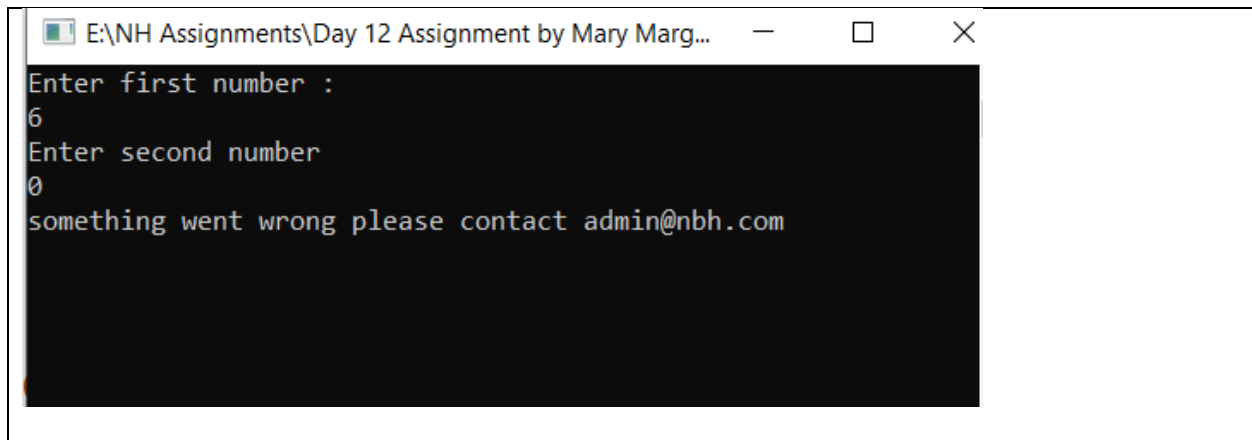
Output 1:



```
E:\NH Assignments\Day 12 A...
Enter first number :
200
Enter second number
100
Division of two numbers : 2
```

Output 2:





```
E:\NH Assignments\Day 12 Assignment by Mary Marg...
Enter first number :
6
Enter second number
0
something went wrong please contact admin@nbh.com
```

Research and write at least 6 exceptions that occur in C# with sample code.

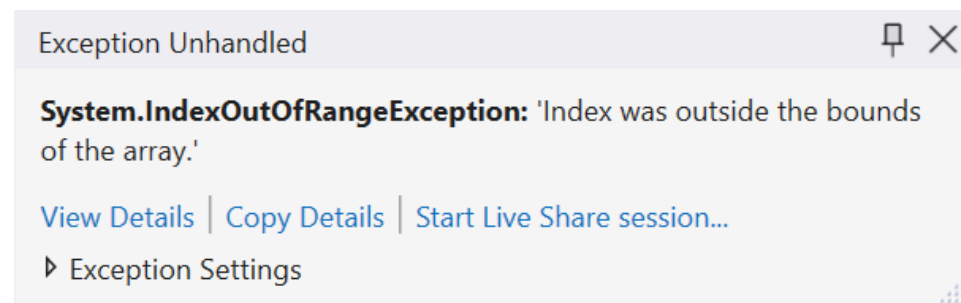
Index Out of Range Exception:

Reason: When you try to access out of range

Code:

```
static void Main (string [] args)
{
    int [] arr = new int [] {1, 2};
    Console.WriteLine(arr[3]);
}
```

Output:



Argument Null Exception:

Reason: When argument in dictionary is initialized as null

Code:

```
static void Main(string[] args)
{
    var dictionary = new Dictionary<string, int>();
    int value = dictionary[null];
}
```

Output:

**System.ArgumentNullException:** 'Value cannot be null.

Parameter name: key'

This exception was originally thrown at this call stack:

[External Code]

Day\_12\_Project\_2.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:

☐ Day 12 Project 2.exe

[Open Exception Settings](#) | [Edit Conditions](#)

Format Exception:

Reason: When we try to give wrong format

Code:

```
static void Main(string[] args)
{
    string data = "Margaret";
    int m=Convert.ToInt32(data);
    Console.WriteLine(m);
}
```

Output:

Exception Unhandled

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

This exception was originally thrown at this call stack:  
[External Code]  
Day\_12\_Project\_3.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:

☐ Day 12 Project 3.exe

Except when thrown from:

[Open Exception Settings](#) | [Edit Conditions](#)

Array Mismatch Exception:

Reason: When we try to attempt value as a type incompatible with array.

Code:

```
static void Main(string[] args)
{
    string[] a = { "Hi" };
    object[] b = a;
    b[0] = 45;
}
```

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:  
Day\_12\_Project\_3.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:

☐ Day 12 Project 3.exe

[Open Exception Settings](#) | [Edit Conditions](#)

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

→ It always executes whether the try block terminates normally or terminates due to an exception.

→ We can run code even if an exception occurs.

→ It doesn't allow any controls to leave the finally block.

#### Example Code:

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

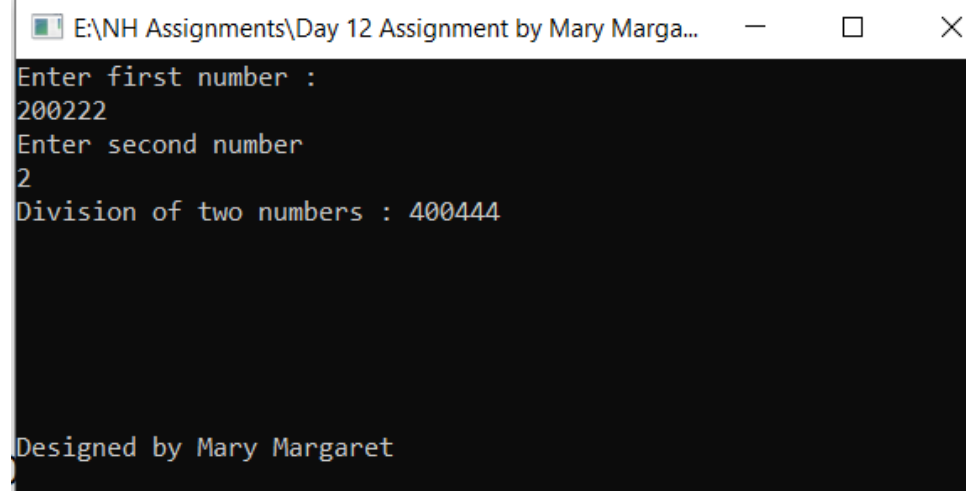
namespace Day_12_Project_2
{
    internal class Program
    {
        //Author : Mary Margaret
        //Example for finally block

        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("Division of two numbers : {0} ", c);
            }
            catch (OverflowException)//exception for large numbers

            {
                Console.WriteLine("please give numbers between 0 and 700000");
                Console.ReadLine();
            }

            finally
            {
                Console.WriteLine("\n\n\n\n\nDesigned by Mary Margaret");
                Console.ReadLine();
            }
        }
    }
}
```

Output:



The screenshot shows a Windows command prompt window with the title bar "E:\NH Assignments\Day 12 Assignment by Mary Marga...". The window contains the following text:

```
Enter first number :  
200222  
Enter second number  
2  
Division of two numbers : 400444  
  
Designed by Mary Margaret
```

Write the 5 points I explained about exception handling.

➤ Syntax:

```
try  
{  
    code  
}  
  
catch (Exception ex)  
{  
}  
finally  
{  
}
```

➤ A single try block can have multiple catch blocks.

➤ We always write General Exception at last.

➤ Statements inside finally will execute every time irrespective of exceptions.

➤ Exception handling is used to handle errors gracefully and display friendly messages.



## Difference Between Compilation Errors and Runtime Errors

Compilation Error	Run time Error
➤ Compilation Error refers to the syntax or semantics in code itself.	➤ Run time Error refers to the error that we get during the execution of code.
➤ We can easily fix Compilation Error.	➤ We face some difficulty in fixing Run time Error.
➤ We can detect it very easily while writing the code.	➤ We cannot detect a runtime error. Because it occurs after total compilation is done.
➤ Ex: Parenthesis, Semicolon etc.	➤ Ex: It occurs due to the division of zero.

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

1. Semicolon Error

```
static void Main(string[] args)
{
    Console.WriteLine()
}
```

2. Using Systems written in comments

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

namespace Day_12_Project_3
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.WriteLine();
        }
    }
}
```

### 3. Case changing Error

```
static void Main(string[] args)
{
    Console.WriteLine();
}
```

### 4. Not Assigning value Error

```
static void Main(string[] args)
{
    int p;
    Console.WriteLine(p);
}
```

#### 5. Duplicate Variables Error

```
static void Main(string[] args)
{
    int p = q;
    p = q;
}
```

#### 6. Assigning wrong data type

```
static void Main(string[] args)
{
    int = "Margaret";
}
```

Write any 6 runtime errors with small code snippets.  
Add run time error screen shots.

#### 1. Format Error

Program.cs

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Day_12_Project_3
8 {
9     0 references
10     internal class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             string data = "Margaret";
16             int m=Convert.ToInt32(data);
17             Console.WriteLine(m);
18         }
19     }
20 }
```

Exception Unhandled

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

This exception was originally thrown at this call stack:

- [External Code]
- Day\_12\_Project\_3.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:
  - ☐ Day 12 Project 3.exe

[Open Exception Settings](#) | [Edit Conditions](#)

121 % No issues found Ln: 20 Ch: 1 SPC CR

## 2. Over Flow Error

Day 12 Project 3

```
9     0 references
10     internal class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             int a, b, c;
16             Console.WriteLine("Enter first number :");
17             a = Convert.ToInt32(Console.ReadLine());
18             Console.WriteLine("Enter second number");
19             b = Convert.ToInt32(Console.ReadLine());
20             c = a / b;
21             Console.WriteLine("Division of two number");
22             Console.ReadLine();
23         }
24     }
25 }
```

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]
- Day\_12\_Project\_3.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:
  - ☐ Day 12 Project 3.exe

[Open Exception Settings](#) | [Edit Conditions](#)

121 % No issues found Ln: 15 Ch: 13 SPC CRU

## 3. Format Error

Day 12 Project 3

0 references

```
internal class Program
{
    0 references
    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("Division of two number");
        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]  
Day\_12\_Project\_3.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:  
☐ Day 12 Project 3.exe

[Open Exception Settings](#) | [Edit Conditions](#)

#### 4. Divide by zero Error

Day 12 Project 3

0 references

```
internal class Program
{
    0 references
    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("Division of two numbers : {0} ". c);
        Console.ReadLine();
    }
}
```

Exception Unhandled

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

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

Exception Settings

5. Mismatch Error

Day 12 Project 3

0 references

```
internal class Program
{
    0 references
    static void Main(string[] args)
    {
        string[] a = { "Hi" };
        object[] b = a;
        b[0] = 45;
    }
}
```

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:  
Day\_12\_Project\_3.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:

☐ Day 12 Project 3.exe

[Open Exception Settings](#) | [Edit Conditions](#)

## 6. Index Out of Range Error

Day 12 Project 3

0 references

```
internal class Program
{
    0 references
    static void Main(string[] args)
    {
        int[] data = new int[3];
        data[10] = 85;
    }
}
```

Exception Unhandled

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

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

Exception Settings