

Assignment 5: Exception Handling in C#

Q1. Handling Division by Zero

Read two numbers and perform division. Use try-catch-finally. Catch DivideByZeroException and display “Division by zero is not allowed.” In the finally block display “Execution completed.” Ensure finally executes regardless of exceptions.

Ans 1. `namespace Assignment_5`

```
{  
    internal class Program  
    {  
        static void Main(string[] args)  
        {  
            Console.WriteLine("Q1. Division Operation:");  
            try  
            {  
                Console.WriteLine("Enter numerator: ");  
                int num1 = Convert.ToInt32(Console.ReadLine());  
  
                Console.WriteLine("Enter denominator: ");  
                int num2 = Convert.ToInt32(Console.ReadLine());  
  
                int result = num1 / num2;  
                Console.WriteLine($"Result = {result}");  
            }  
            catch (DivideByZeroException)  
            {  
                Console.WriteLine("Division by zero is not allowed.");  
            }  
            finally  
            {  
                Console.WriteLine("Execution completed.\n");  
                Console.WriteLine("Developed by Deepanshu Dixit.");  
            }  
        }  
    }  
}
```

Output:

```
Q1. Division Operation:  
Enter numerator:  
10  
Enter denominator:  
2  
Result = 5  
Execution completed.  
  
Developed by Deepanshu Dixit.  
  
C:\Users\admin\source\repos\Assignment 5.cs\bin\Debug\net8.0\Assignment 5.cs.exe (process 5116) exited with code 0 (0x0).  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .
```

```
Q1. Division Operation:  
Enter numerator:  
10  
Enter denominator:  
0  
Division by zero is not allowed.  
Execution completed.  
  
Developed by Deepanshu Dixit.  
  
C:\Users\admin\source\repos\Assignment 5.cs\bin\Debug\net8.0\Assignment 5.cs.exe (process 6524) exited with code 0 (0x0)  
. . .  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .
```

Q2. Multiple Catch Blocks

Read console input and convert to int. Handle FormatException, OverflowException, and a generic Exception, with distinct messages.

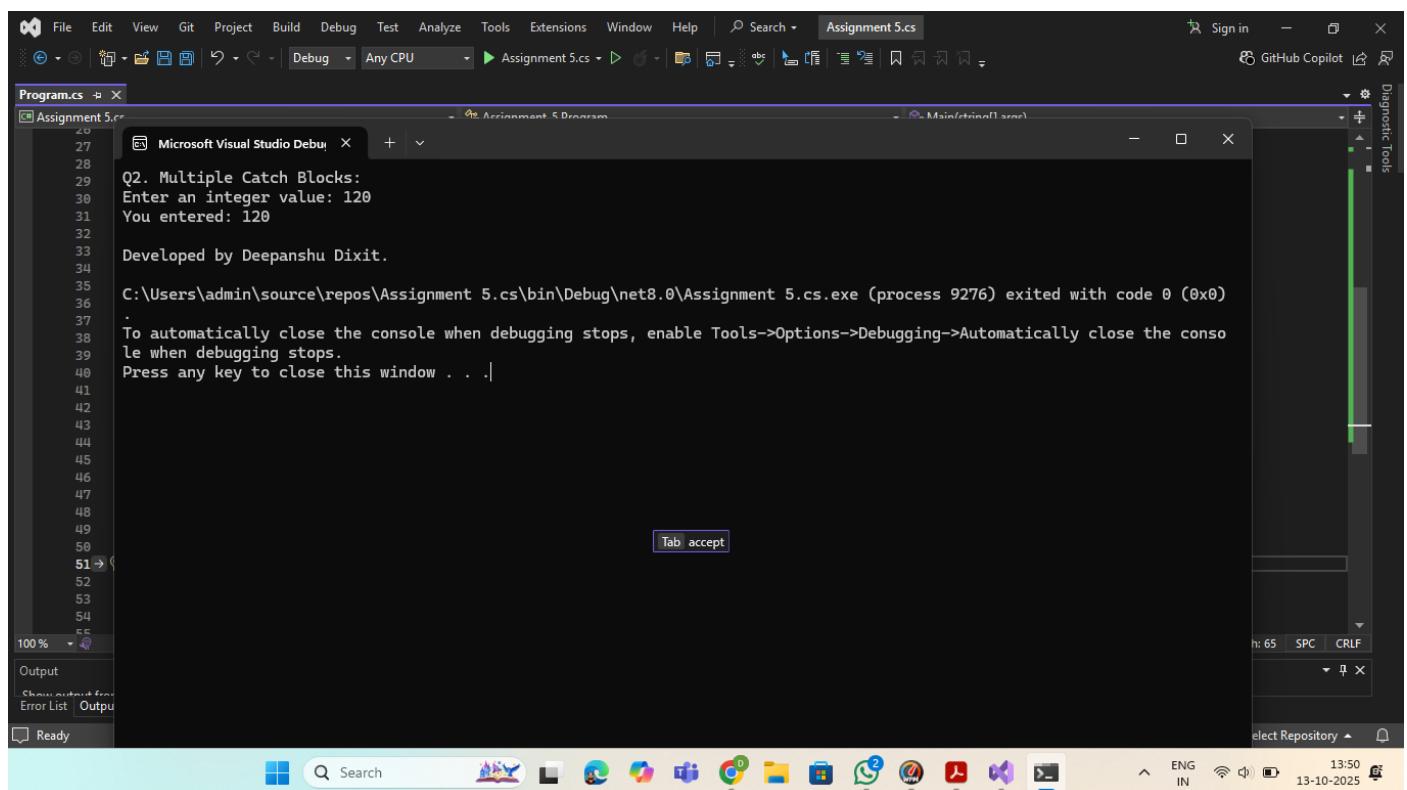
Ans 2. `namespace Assignment_5`
{
 `internal class Program`

```

{
    static void Main(string[] args)
    {
        Console.WriteLine("Q2. Multiple Catch Blocks:");
        Console.Write("Enter an integer value: ");
        try
        {
            int value = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine($"You entered: {value}");
        }
        catch (FormatException)
        {
            Console.WriteLine("Invalid format. Please enter a numeric value.");
        }
        catch (OverflowException)
        {
            Console.WriteLine("Number too large or too small for an Int32.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"An unexpected error occurred: {ex.Message}");
        }
        Console.WriteLine();
        Console.WriteLine("Developed by Deepanshu Dixit.");
    }
}

```

Output:



The screenshot shows the Microsoft Visual Studio interface with the 'Assignment 5.cs' file open. The output window displays the following text:

```

27
28 Q2. Multiple Catch Blocks:
29 Enter an integer value: 120
30 You entered: 120
31
32
33 Developed by Deepanshu Dixit.
34
35 C:\Users\admin\source\repos\Assignment 5.cs\bin\Debug\net8.0\Assignment 5.cs.exe (process 9276) exited with code 0 (0x0)
36 .
37 To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
38 le when debugging stops.
39 Press any key to close this window . . .
40
41
42
43
44
45
46
47
48
49
50
51 → (
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 %

```

The status bar at the bottom right shows the date and time as 13-10-2025 13:50.

```
Q2. Multiple Catch Blocks:  
Enter an integer value: 120.57  
Invalid format. Please enter a numeric value.  
Developed by Deepanshu Dixit.  
C:\Users\admin\source\repos\Assignment 5.cs\bin\Debug\net8.0\Assignment 5.cs.exe (process 9700) exited with code 0 (0x0)  
. . .  
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.  
Press any key to close this window . . .
```

Q3. Custom Exception — NegativeSalaryException

Define NegativeSalaryException : Exception. If entered salary < 0, throw it and handle with a clear error message.

Ans 3. `using System;`

```
public class NegativeSalaryException : Exception
{
    public NegativeSalaryException(string message) : base(message)
    {
    }
}
class Program
{
    static void Main()
    {
        try
        {
            Console.Write("Enter your salary: ");

            double salary = Convert.ToDouble(Console.ReadLine());

            if (salary < 0)
            {
                throw new NegativeSalaryException("Salary cannot be negative.");
            }
            Console.WriteLine($"Salary entered: {salary}");
        }
        catch (NegativeSalaryException ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
        catch (FormatException)
        {
            Console.WriteLine("Invalid input. Please enter a numeric salary value.");
        }
    }
}
```

```

        }
    finally
    {
        Console.WriteLine("Execution completed.");
    }
    Console.WriteLine("Developed by Deepanshu Dixit.");
}
}

```

Output:

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a GitHub Copilot icon. The title bar says "ConsoleApp9". The main window displays the code in "Program.cs" and the output from the debugger. The output window shows the following text:

```

32
33
34
35 Enter your salary: 150000
36 Salary entered: 150000
37 Execution completed.
38 Developed by Deepanshu Dixit.
39
40 C:\Users\admin\source\repos\ConsoleApp9\bin\Debug\net8.0\ConsoleApp9.exe (process 14476) exited with code 0 (0x0).
41 To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
42 le when debugging stops.
43 Press any key to close this window . . .

```

The status bar at the bottom indicates "14:10 13-10-2025".

This screenshot shows the same Microsoft Visual Studio environment as the previous one. The output window displays the following text:

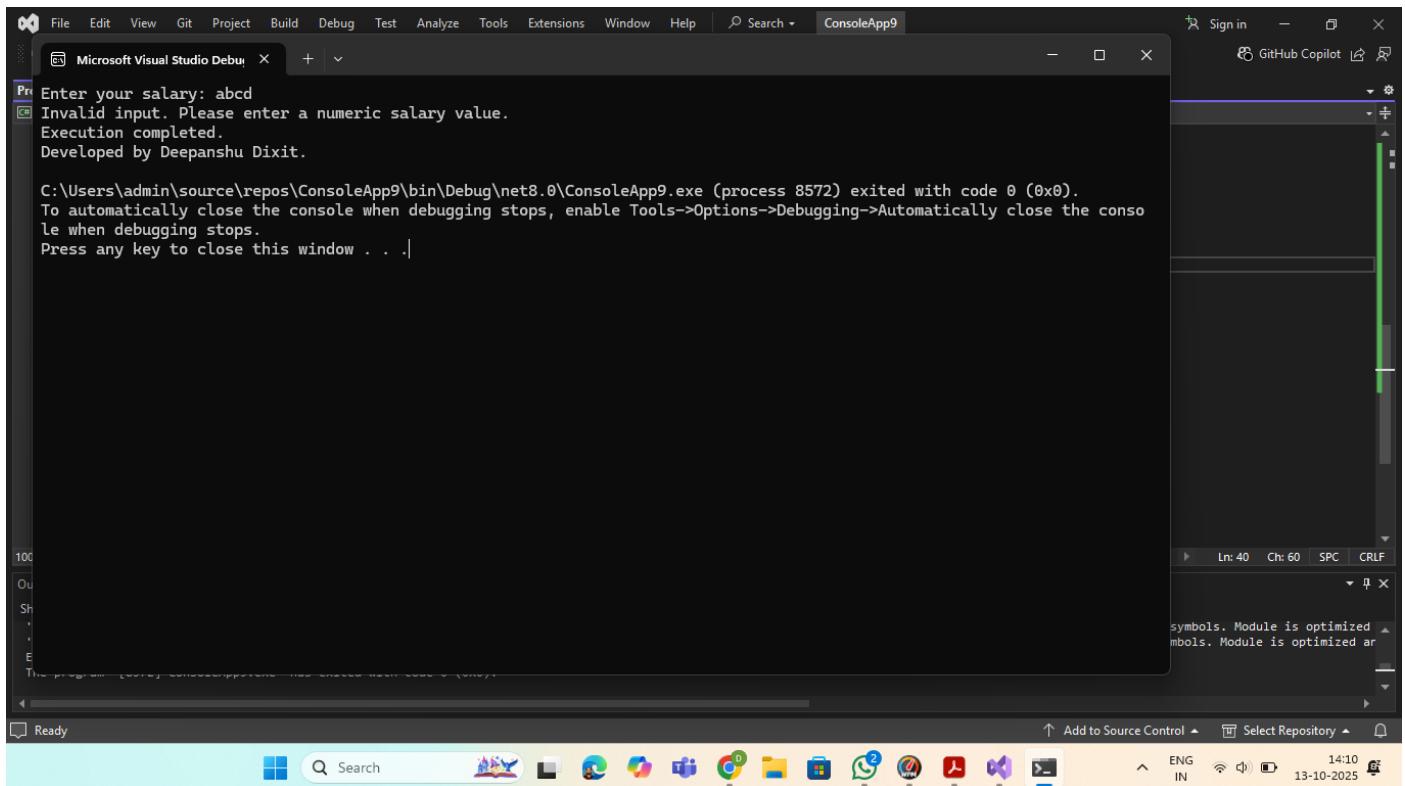
```

Enter your salary: -150000
Error: Salary cannot be negative.
Execution completed.
Developed by Deepanshu Dixit.

C:\Users\admin\source\repos\ConsoleApp9\bin\Debug\net8.0\ConsoleApp9.exe (process 8432) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .

```

The status bar at the bottom indicates "14:10 13-10-2025".



Q4. Banking Scenario — InsufficientBalanceException

Simulate withdrawal: if withdrawal > balance, throw custom InsufficientBalanceException; otherwise print remaining balance.

Ans 4. using System;

```
public class InsufficientBalanceException : Exception
{
    public InsufficientBalanceException(string message) : base(message)
    {
    }
}

class Program
{
    static void Main()
    {
        double balance = 5000.0;
        Console.WriteLine($"Your current balance: ₹{balance}");

        try
        {
            Console.Write("Enter withdrawal amount: ");

            double withdraw = Convert.ToDouble(Console.ReadLine());

            if (withdraw > balance)
            {
                throw new InsufficientBalanceException("Insufficient balance for this
withdrawal.");
            }
            balance -= withdraw;
            Console.WriteLine($"Withdrawal successful! Remaining balance: ₹{balance}");
        }
        catch (InsufficientBalanceException ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
        catch (FormatException)
```

```

    {
        Console.WriteLine("Invalid input! Please enter a valid number.");
    }
    finally
    {
        Console.WriteLine("Transaction completed.");
    }
    Console.WriteLine("Developed by Deepanshu Dixit.");
}

```

Output:

```

Microsoft Visual Studio Debug x + v
Your current balance: 75000
Enter withdrawal amount: 2000
Withdrawal successful! Remaining balance: 73000
Transaction completed.
Developed by Deepanshu Dixit.

C:\Users\admin\source/repos\ConsoleApp9\bin\Debug\net8.0\ConsoleApp9.exe (process 2764) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

```

Microsoft Visual Studio Debug x + v
Your current balance: 75000
Enter withdrawal amount: 7000
Error: Insufficient balance for this withdrawal.
Transaction completed.
Developed by Deepanshu Dixit.

C:\Users\admin\source/repos\ConsoleApp9\bin\Debug\net8.0\ConsoleApp9.exe (process 17356) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

Q5. Student Marks Validation

Student class stores marks (0–100). If input outside range, throw InvalidMarksException. Demonstrate validation and handling in Main().

```
Ans 5. using System;
public class InvalidMarksException : Exception
{
    public InvalidMarksException(string message) : base(message)
    {
    }
}

public class Student
{
    private int marks;

    public int Marks
    {
        get { return marks; }
        set
        {
            if (value < 0 || value > 100)
                throw new InvalidMarksException("Marks must be between 0 and 100.");
            marks = value;
        }
    }

    public Student(int marks)
    {
        Marks = marks;
    }
}

class Program
{
    static void Main()
    {
        try
        {
            Console.Write("Enter student marks (0-100): ");

            int inputMarks = Convert.ToInt32(Console.ReadLine());

            Student s = new Student(inputMarks);

            Console.WriteLine($"Valid marks entered: {s.Marks}");
        }
        catch (InvalidMarksException ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
        catch (FormatException)
        {
            Console.WriteLine("Invalid input! Please enter numeric marks.");
        }
        catch (OverflowException)
        {
            Console.WriteLine("Number too large or too small for an integer.");
        }
        finally
        {
            Console.WriteLine("Marks validation completed.");
        }
        Console.WriteLine("Developed by Deepanshu Dixit.");
    }
}
```

Output:

The screenshot shows the Microsoft Visual Studio interface with the 'ConsoleApp9' project selected. The output window displays the following text:

```
Enter student marks (0-100): 95
Valid marks entered: 95
Marks validation completed.
Developed by Deepanshu Dixit.

C:\Users\admin\source\repos\ConsoleApp9\bin\Debug\net8.0\ConsoleApp9.exe (process 7944) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```

The status bar at the bottom right shows the date and time as 13-10-2025 14:52.

The screenshot shows the Microsoft Visual Studio interface with the 'ConsoleApp9' project selected. The output window displays the following text:

```
Enter student marks (0-100): 110
Error: Marks must be between 0 and 100.
Marks validation completed.
Developed by Deepanshu Dixit.

C:\Users\admin\source\repos\ConsoleApp9\bin\Debug\net8.0\ConsoleApp9.exe (process 6584) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```

The status bar at the bottom right shows the date and time as 13-10-2025 14:52.

