

Lab Assignment 10: Development of C# Console Applications

INTRODUCTION, SETUP, AND TOOLS

Introduction

This lab is designed to introduce you to the fundamental concepts of software development using C# and the .NET framework. The primary focus is on creating console-based applications to gain hands-on experience with the basic syntax, control structures, and object-oriented programming (OOP) principles in C#.

Setup

we need to have

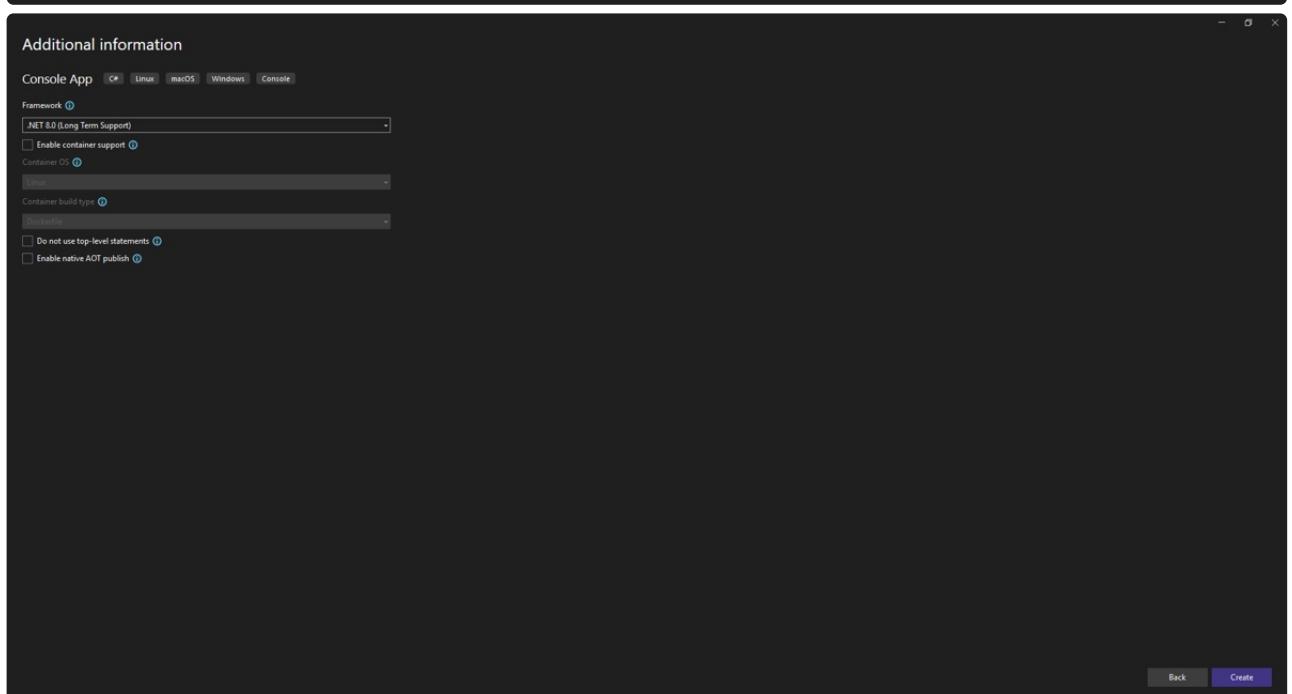
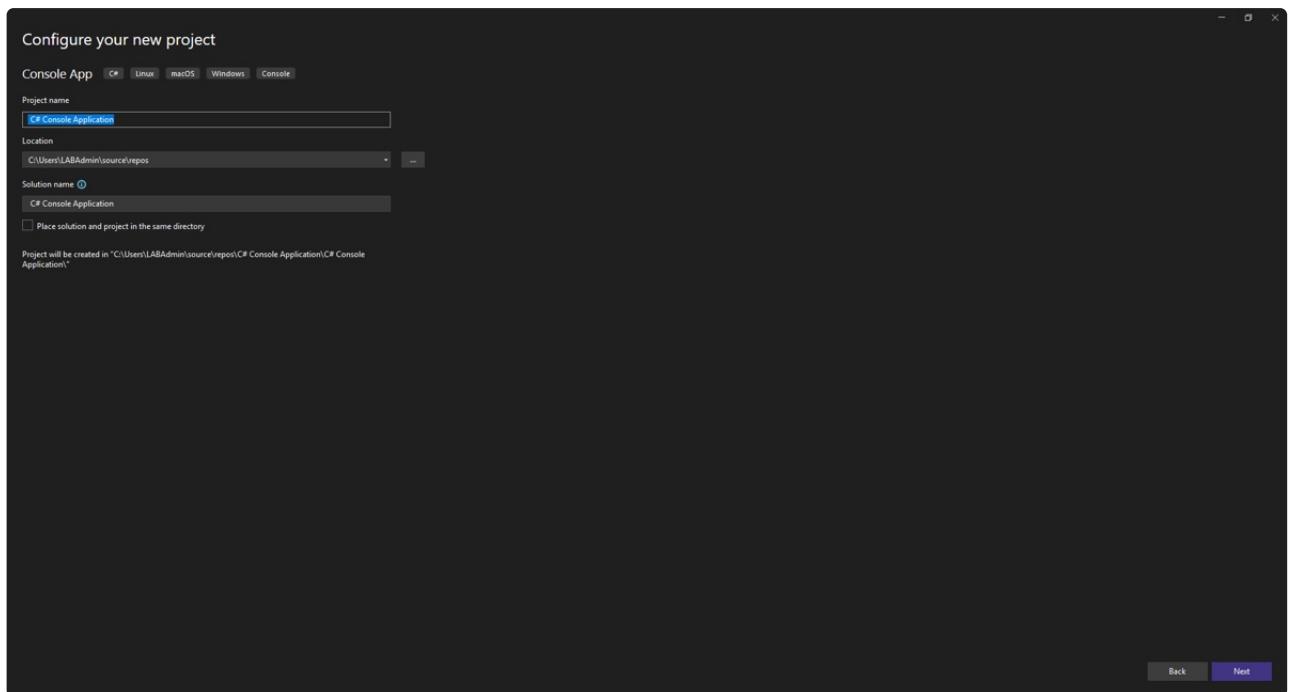
- **Visual Studio 2022 (Community Edition)** – This is the IDE where you'll be writing and executing your C# programs.
- **.NET SDK** – The Software Development Kit (SDK) for the .NET framework should be installed to ensure compatibility with the latest versions of C#.

Tools

- **.NET SDK:**
 - The .NET SDK provides the necessary libraries and tools to develop, run, and deploy .NET applications. You'll use this to build and execute your C# console applications.
- **C# Programming Language:**
 - C# is an object-oriented programming language used to develop a wide range of applications. In this lab, you will learn how to use its syntax, control structures (loops, conditionals), and object-oriented principles (classes, inheritance, etc.).

SETTING UP .NET DEVELOPMENT ENVIRONMENT

Open Visual Studio and create a new C# Console Application project. with .NET target framework be 6 or later.



```
using System;
class Program
{
    static void Main()
    {
        Console.WriteLine("Its Me!");
    }
}
```

```
C:\> C# Console Application.exe
Its Me!
The program '[28804] C# Console Application.exe' has exited with code 0 (0x0).
```

Running simple program:

(prints "Its, Me!" in the console)

```
C:\> C# Console Application.exe
Its Me!
The program '[28844] C# Console Application.exe' has exited with code 0 (0x0).
```

UNDERSTANDING BASIC SYNTAX AND CONTROL STRUCTURES



Input: 0 and 5

expected output: Sum: 5 (odd), Difference: -5, Product: 0, Division: 0 (since 0 / 5 = 0)

The screenshot shows the Microsoft Visual Studio interface with the 'Debug' window on the left displaying the program's output:

```
Enter the first number: 0
Enter the second number: 5
Sum: 5
Difference: -5
Product: 0
Division: 0
The sum is odd.

C:\Users\LABAdmin\source\repos\Console Application\Console Application\bin\Debug\net8.0\Console Application.exe (process 15748) 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 right side of the screen shows the code editor with the 'Calculator.cs' file open. The code defines a class 'Calculator' with methods for input, performing operations, and outputting results based on the parity of the sum.

```
class Calculator
{
    private int number1;
    private int number2;

    public void GetInput()
    {
        Console.Write("Enter the first number: ");
        number1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Enter the second number: ");
        number2 = Convert.ToInt32(Console.ReadLine());
    }

    public void PerformOperations()
    {
        int sum = number1 + number2;
        int difference = number1 - number2;
        int product = number1 * number2;
        double division = number2 != 0 ? (double)number1 / number2 : 0;

        Console.WriteLine($"Sum: {sum}");
        Console.WriteLine($"Difference: {difference}");
        Console.WriteLine($"Product: {product}");

        if (number2 != 0)
            Console.WriteLine($"Division: {division}");
        else
            Console.WriteLine("Division by zero is not allowed.");

        if (sum % 2 == 0)
            Console.WriteLine("The sum is even.");
        else
            Console.WriteLine("The sum is odd.");
    }
}
```

Input: -3 and -7

expected output: Sum: -10 (even), Difference: 4, Product: 21, Division: 0.4286 (since -3 / -7 = 0.4286)

The screenshot shows the Microsoft Visual Studio interface with the 'Debug' window on the left displaying the program's output:

```
Enter the first number: -3
Enter the second number: -7
Sum: -10
Difference: 4
Product: 21
Division: 0.42857142857142855
The sum is even.

C:\Users\LABAdmin\source\repos\Console Application\Console Application\bin\Debug\net8.0\Console Application.exe (process 26212) 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 right side of the screen shows the code editor with the 'Calculator.cs' file open. The code is identical to the previous screenshot, handling input and output for negative numbers.

```
class Calculator
{
    private int number1;
    private int number2;

    public void GetInput()
    {
        Console.Write("Enter the first number: ");
        number1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Enter the second number: ");
        number2 = Convert.ToInt32(Console.ReadLine());
    }

    public void PerformOperations()
    {
        int sum = number1 + number2;
        int difference = number1 - number2;
        int product = number1 * number2;
        double division = number2 != 0 ? (double)number1 / number2 : 0;

        Console.WriteLine($"Sum: {sum}");
        Console.WriteLine($"Difference: {difference}");
        Console.WriteLine($"Product: {product}");

        if (number2 != 0)
            Console.WriteLine($"Division: {division}");
        else
            Console.WriteLine("Division by zero is not allowed.");

        if (sum % 2 == 0)
            Console.WriteLine("The sum is even.");
        else
            Console.WriteLine("The sum is odd.");
    }
}
```

Input: 5.5 and 3.2

expected output: The program will not run

```
using System;
class Calculator
{
    private int number1;
    private int number2;

    public void GetInput()
    {
        Console.Write("Enter the first number: ");
        while (!int.TryParse(Console.ReadLine(), out number1))
        {
            Console.WriteLine("Error: Invalid input! Please enter an integer.");
            Console.Write("Enter the first number: ");
        }

        Console.Write("Enter the second number: ");
        while (!int.TryParse(Console.ReadLine(), out number2))
        {
            Console.WriteLine("Error: Invalid input! Please enter an integer.");
            Console.Write("Enter the second number: ");
        }
    }

    public void PerformOperations()
    {
        int sum = number1 + number2;
        int difference = number1 - number2;
        int product = number1 * number2;

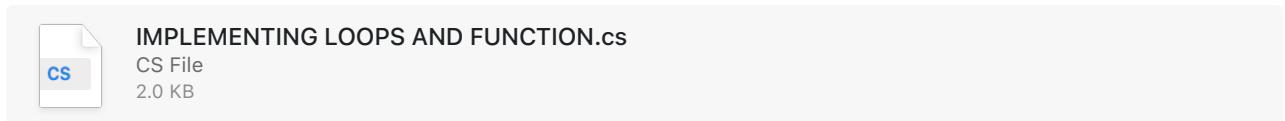
        // Handle division by zero
        if (number2 != 0)
        {
            double division = (double)number1 / number2;
            Console.WriteLine($"Division: {division}");
        }
    }
}

No issues found
```

Output

```
'C# Console Application.exe' (CoreCLR: cihost): Loaded 'C:\V...'.
```

IMPLEMENTING LOOPS AND FUNCTION



Input: 7, 1, -5, 1.1, 20

exit to end code

expected output: 1, 1, Invalid, Invalid, 243290200817664000

```

Numbers from 1 to 10:
1
2
3
4
5
6
7
8
9
10
Enter a number to calculate its factorial or type 'exit' to quit:
Input: 0
Factorial of 0 is 1
Input: 1
Factorial of 1 is 1
Input: -5
Invalid input! Please enter a non-negative integer or 'exit' to quit.
Input: 1.1
Invalid input! Please enter a non-negative integer or 'exit' to quit.
Input: 20
Factorial of 20 is 2432902008176640000
Input: exit
Exiting the program. Goodbye!

```

C:\Users\LABAdmin\source/repos\C# Console Application\bin\Debug\net8.0\C# Console Application.exe (process 28212) 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 . . .

PERFORM OBJECT-ORIENTED PROGRAMMING IN C#

PERFORM OBJECT-ORIENTED PROGRAMMING.cs
CS File
2.0 KB

```

== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

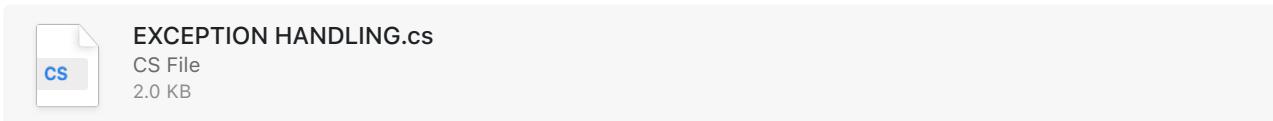
Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B
Hostel Name (IITGN): G Hostel

```

C:\Users\LABAdmin\source/repos\C# Console Application\bin\Debug\net8.0\C# Console Application.exe (process 4572) 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 . . .

EXCEPTION HANDLING



Input: 0 and 1

expected output: Invalid or Infinity

```
using System;
class Calculator
{
    private int number1;
    private int number2;

    public void GetInput()
    {
        Console.WriteLine("Enter the first number: ");
        while (true)
        {
            try
            {
                number1 = int.Parse(Console.ReadLine());
                break; // Exit loop if input is valid
            }
            catch (FormatException)
            {
                Console.WriteLine("Error: Invalid input! Please enter an integer.");
                Console.WriteLine("Enter the first number: ");
            }
        }

        Console.WriteLine("Enter the second number: ");
        while (true)
        {
            try
            {
                number2 = int.Parse(Console.ReadLine());
                break; // Exit loop if input is valid
            }
            catch (FormatException)
            {
                Console.WriteLine("Error: Invalid input! Please enter an integer.");
                Console.WriteLine("Enter the second number: ");
            }
        }
    }
}
```

DEBUGGING USING VISUAL STUDIO DEBUGGER

For Activities (2)

Enter the first number: 5
Enter the second number: 6

```
Program.cs  C# Console Application  Main(string[] args)
```

```
class Program
{
    static void Main(string[] args)
    {
        Calculator calc = new Calculator();
        calc.GetInput();
        calc.PerformOperations();
    }
}
```

Autos Locals Watch1 Error List Call Stack Breakpoi... Exception... Command... Immediate... Output Error List

Enter the first number: 5
Enter the second number: 6

```
Calculator.cs  C# Console Application  PerformOperations()
```

```
public void PerformOperations()
{
    if (number1 < 0 || number2 < 0)
    {
        Console.WriteLine("Invalid input! Please enter an integer.");
        Console.WriteLine("Enter the second number: ");
    }
    else
    {
        int sum = number1 + number2;
        int difference = number1 - number2;
        int product = number1 * number2 - 1;

        // Handle division by zero
        if (number2 != 0)
        {
            double division = (double)number1 / number2;
            Console.WriteLine($"Division: {division}");
        }
    }
}
```

Autos Locals Watch1 Error List Call Stack Breakpoi... Exception... Command... Immediate... Output Error List

```
Enter the first number: 5
Enter the second number: 6

using System;

namespace CSharp_Console_Application
{
    class Calculator
    {
        public void PerformOperations()
        {
            int sum = number1 + number2;
            int difference = number1 - number2;
            int product = number1 * number2; // 30ms elapsed

            // Handle division by zero
            if (number2 != 0)
            {
                double division = (double)number1 / number2;
                Console.WriteLine($"Division: {division}");
            }
        }
    }
}

1 reference
public void PerformOperations()
{
    int sum = number1 + number2;
    int difference = number1 - number2;
    int product = number1 * number2; // 30ms elapsed

    // Handle division by zero
    if (number2 != 0)
    {
        double division = (double)number1 / number2;
        Console.WriteLine($"Division: {division}");
    }
}

No issues found
```

Autos Locals Watch1 Error List

Search (Ctrl+E) Entire Solution 0 Errors 0 Warnings

Name Value Type

difference	0	int
number1	5	int
number2	6	int
sum	11	int
this	(Calculator)	Calculator

Call Stack Breakpoi... Exception... Command... Immediate... Output Error List

```
Enter the first number: 5
Enter the second number: 6

using System;

namespace CSharp_Console_Application
{
    class Calculator
    {
        public void PerformOperations()
        {
            int sum = number1 + number2;
            int difference = number1 - number2;
            int product = number1 * number2; // 30ms elapsed

            // Handle division by zero
            if (number2 != 0)
            {
                double division = (double)number1 / number2;
                Console.WriteLine($"Division: {division}");
            }
        }
    }
}

1 reference
public void PerformOperations()
{
    int sum = number1 + number2;
    int difference = number1 - number2;
    int product = number1 * number2; // 30ms elapsed

    // Handle division by zero
    if (number2 != 0)
    {
        double division = (double)number1 / number2;
        Console.WriteLine($"Division: {division}");
    }
}

No issues found
```

Autos Locals Watch1 Error List

Search (Ctrl+E) Entire Solution 0 Errors 0 Warnings

Name Value Type

difference	-1	int
number1	5	int
number2	0	int
product	0	int
this	(Calculator)	Calculator

Call Stack Breakpoi... Exception... Command... Immediate... Output Error List

```
Enter the first number: 5
Enter the second number: 6

public void PerformOperations()
{
    int sum = number1 + number2;
    int difference = number1 - number2;
    int product = number1 * number2;

    // Handle division by zero
    if (number2 != 0)
    {
        double division = (double)number1 / number2;
        Console.WriteLine($"Division: {division}");
    }
}

Name      Value   Type
---      ---   ---
number1      5   int
number2      6   int
product     30  int
this       (Calculator)
```



```
Enter the first number: 5
Enter the second number: 6
Division: 0.8333333333333334
Sum: 11
Difference: -1
Product: 30

Name      Value   Type
---      ---   ---
sum          11  int
this       (Calculator)
```

The screenshot shows the Microsoft Visual Studio interface with a console application running. The output window displays:

```
Enter the first number: 5
Enter the second number: 6
Division: 0.8333333333333334
Sum: 11
Difference: -1
Product: 30
The sum is odd.
```

The code in Program.cs is:

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

namespace C# Console Application
{
    class Program
    {
        static void Main(string[] args)
        {
            Calculator calc = new Calculator();
            calc.GetInput();
            calc.PerformOperations();
        }
    }

    class Calculator
    {
        public void GetInput()
        {
            int num1 = 5;
            int num2 = 6;
        }

        public void PerformOperations()
        {
            double division = num1 / num2;
            double sum = num1 + num2;
            int difference = num1 - num2;
            int product = num1 * num2;

            Console.WriteLine("Division by zero is not allowed.");
            Console.WriteLine($"Sum: {sum}");
            Console.WriteLine($"Difference: {difference}");
            Console.WriteLine($"Product: {product}");

            // Check if the sum is even or odd
            if (sum % 2 == 0)
                Console.WriteLine("The sum is even.");
            else
                Console.WriteLine("The sum is odd.");
        }
    }
}
```

The output window shows the application's output and some internal process information. The status bar at the bottom indicates it's 35°C in Spokane, Washington, and the date is 07.04.2024.

For Activities (3)

The screenshot shows the Microsoft Visual Studio interface with a console application running. The output window displays:

```
Process: [31564] C# Console Application.exe -> Lifecycle Events -> Thread: [24004] Main Thread
```

The code in Program.cs is:

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

namespace C# Console Application
{
    class Program
    {
        static void Main(string[] args)
        {
            NumberOperations operations = new NumberOperations();

            // Calling the methods
            operations.PrintNumbers();
            operations.KeepAskingForInput();
        }
    }

    class NumberOperations
    {
        public void PrintNumbers()
        {
            int num1 = 5;
            int num2 = 6;
        }

        public void KeepAskingForInput()
        {
            int num1 = 5;
            int num2 = 6;
        }
    }
}
```

The Error List window shows two warnings:

Name	Type	Description	Project	File
operations	NumberOperations	CS9600: Converting null literal or possible null value to non nullable type.	C# Console Application	Program.cs
operations	NumberOperations	CS8602: Dereference of a possibly null reference.	C# Console Application	Program.cs

The status bar at the bottom indicates it's 35°C in Spokane, Washington, and the date is 07.04.2024.

```
Numbers from 1 to 10:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

```
class NumberOperations  
{  
    // Method to print numbers from 1 to 10 using a for loop  
    public void PrintNumbers()  
    {  
        Console.WriteLine("Numbers from 1 to 10:");  
        for (int i = 1; i <= 10; i++)  
        {  
            Console.WriteLine(i);  
        }  
    }  
  
    // Method to calculate the factorial of a number  
    public long CalculateFactorial(int number)  
    {  
        long factorial = 1;  
        for (int i = 1; i <= number; i++)  
        {  
            factorial *= i;  
        }  
        return factorial;  
    }  
}
```

```
class Program  
{  
    static void Main(string[] args)  
    {  
        NumberOperations operations = new NumberOperations();  
        operations.PrintNumbers();  
        long factorial = operations.CalculateFactorial(5);  
        Console.WriteLine($"Factorial of 5 is: {factorial}");  
    }  
}
```

The image displays two side-by-side screenshots of the Microsoft Visual Studio IDE interface, showing the execution of a C# console application.

Top Screenshot:

- Code:** Shows the `Program.cs` file with the following code:

```
Numbers from 1 to 10:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

```
Class Program  
{  
    static void Main(string[] args)  
    {  
        NumberOperations operations = new NumberOperations();  
  
        // Calling the methods  
        operations.PrintNumbers();  
        operations.KeepAskingForInput();  
    }  
}
```
- Output Window:** Displays the output "Numbers from 1 to 10:" followed by the numbers 1 through 10.
- Watch Window:** Shows the variable `operations` of type `NumberOperations`.
- Error List:** Shows two warnings:
 - CS8600: Converting null literal or possible null value to non-nullable type.
 - CS8602: Dereference of a possibly null reference.

Bottom Screenshot:

- Code:** Shows the `Program.cs` file with the following code:

```
Numbers from 1 to 10:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Enter a number to calculate its factorial or type 'exit' to quit:  
Input: 20
```

```
Class Program  
{  
    static void Main(string[] args)  
    {  
        string input;  
        Console.WriteLine("Enter a number to calculate its factorial or type 'exit' to quit");  
  
        while (true)  
        {  
            Console.Write("Input: ");  
            input = Console.ReadLine();  
  
            if (input.ToLower() == "exit")  
            {  
                Console.WriteLine("Exiting the program. Goodbye!");  
                break;  
            }  
  
            if (int.TryParse(input, out int number) && number >= 0)  
            {  
                long result = CalculateFactorial(number);  
                Console.WriteLine($"Factorial of {number} is {result}");  
            }  
        }  
    }  
}
```
- Output Window:** Displays the output "Numbers from 1 to 10:" followed by the numbers 1 through 10, then "Enter a number to calculate its factorial or type 'exit' to quit:", and finally "Input: 20".
- Watch Window:** Shows the variables `input` (string) and `this` (NumberOperations).
- Error List:** Shows two warnings:
 - CS8600: Converting null literal or possible null value to non-nullable type.
 - CS8602: Dereference of a possibly null reference.

The image displays two side-by-side screenshots of the Microsoft Visual Studio IDE interface, version 2022, running on a Windows operating system.

Top Screenshot:

- Code Editor:** Shows the `Program.cs` file for a "C# Console Application". The code implements a loop to keep asking for input until the user types "exit". It then attempts to parse the input as an integer to calculate its factorial using a helper method.
- Output Window:** Displays the application's output, showing the numbers from 1 to 10 followed by a prompt for user input.
- Watch Window:** Shows the variable state with `input` set to "20", `number` set to 0, and `this` pointing to an instance of `NumberOperations`.
- Error List:** Shows two warnings: CS8600 (Converting null literal or possible null value to non-nullable type) and CS8602 (Dereference of a possibly null reference).

Bottom Screenshot:

- Code Editor:** Shows the same `Program.cs` file, but the code has been refactored. The factorial calculation is moved into a separate method, `CalculateFactorial(int number)`, which uses a for loop to calculate the factorial of `number`. The original `KeepAskingForInput()` method remains, but it is now empty.
- Watch Window:** Shows the variable state with `factorial` set to 1, `i` set to 1, `number` set to 20, and `this` pointing to an instance of `NumberOperations`.
- Error List:** Shows the same two warnings as the top screenshot.

Numbers from 1 to 10:
1
2
3
4
5
6
7
8
9
10
Enter a number to calculate its factorial or type 'exit' to quit:
Input: 20

```
1  // Method to calculate the factorial of a number
2  public long CalculateFactorial(int number)
3  {
4      long factorial = 1;
5      for (int i = 1; i <= number; i++)
6      {
7          factorial *= i; // 1ms elapsed
8      }
9      return factorial;
10 }
11
12 // Method to handle user input until they enter 'exit'
13 public void KeepAskingForInput()
14 {
15     string input;
16 }
```

Autos

Name	Value	Type
factorial	40320	long
i	9	int
number	20	int
this	(NumberOperations)	NumberO...

Error List

Code	Description	Project	File
CS8600	Converting null literal or possible null value to non nullable type.	C# Console Application	Program.cs
CS8602	Dereference of a possibly null reference.	C# Console Application	Program.cs

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Numbers from 1 to 10:

```
1
2
3
4
5
6
7
8
9
10
```

Enter a number to calculate its factorial or type 'exit' to quit:

Input: 20

```
1
2
3
4
5
6
7
8
9
10
```

Numbers from 1 to 10:

```
1
2
3
4
5
6
7
8
9
10
```

Enter a number to calculate its factorial or type 'exit' to quit:

Input: 20

```
1
2
3
4
5
6
7
8
9
10
```

```
Numbers from 1 to 10:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Enter a number to calculate its factorial or type 'exit' to quit:  
Input: 20  
Factorial of 20 is 2432902008176640000  
Input: |
```

```
Program.cs  C# Console Application - NumberOperations - CalculateFactorial(int number)  
    }  
    // Method to calculate the factorial of a number  
    public long CalculateFactorial(int number)  
    {  
        long factorial = 1;  
        for (int i = 1; i <= number; i++)  
        {  
            factorial *= i;  
        }  
        return factorial;  
    }  
  
    // Method to handle user input until they enter 'exit'  
    public void KeepAskingForInput()  
    {  
        string input;  
        while (true)  
        {  
            Console.WriteLine("Enter a number to calculate its factorial or type 'exit' to quit:  
Input: 20  
Factorial of 20 is 2432902008176640000  
Input: exit
```

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.

```
Numbers from 1 to 10:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Enter a number to calculate its factorial or type 'exit' to quit:  
Input: 20  
Factorial of 20 is 2432902008176640000  
Input: exit
```

```
Program.cs  C# Console Application - NumberOperations - KeepAskingForInput()  
    }  
    // Method to calculate the factorial of a number  
    public long CalculateFactorial(int number)  
    {  
        long result = CalculateFactorial(number);  
        Console.WriteLine($"Factorial of {number} is {result}");  
    }  
  
    // Method to handle user input until they enter 'exit'  
    public void KeepAskingForInput()  
    {  
        string input;  
        Console.WriteLine("Enter a number to calculate its factorial or type 'exit' to quit:  
Input: 20  
Factorial of 20 is 2432902008176640000  
Input: exit
```

Snipping Tool
Screenshot copied to clipboard
Automatically saved to screenshots folder.

```
Numbers from 1 to 10:  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
Enter a number to calculate its factorial or type 'exit' to quit:  
Input: 28  
Factorial of 28 is 2432902880176640000  
Input: exit  
Exiting the program. Goodbye!  
C:\Users\LABAdmin\source\repos\C# Console Application\C# Console Application\bin\Debug\net8.0\C# Console Application.exe (process 31564) 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 screenshot shows the Microsoft Visual Studio interface with a C# console application project open. The code in Program.cs handles user input for calculating factorials or exiting the program. A screenshot tool window is visible on the right, showing a preview of the desktop.

For Activities (4)

```
public StudentIITGN(string name, int id, int marks, string hostelName)  
    : base(name, id, marks)  
    {  
        Hostel_Name_IITGN = hostelName;  
    }  
  
    // Override DisplayDetails method to include hostel info  
    public override void DisplayDetails()  
    {  
        base.DisplayDetails();  
        Console.WriteLine($"Hostel Name (IITGN): {Hostel_Name_IITGN}");  
    }  
  
    static void Main(string[] args)  
    {  
        // Creating a Student object  
        Student student1 = new Student("Vipul Sunil Patil", 22110189, 88);  
        Student student2 = new Student("Ranit Bicash", 22110217, 85);  
    }
```

The screenshot shows the Microsoft Visual Studio interface with a C# console application project open. The code creates Student objects and prints their details, including the specific hostel name for each student. A screenshot tool window is visible on the right, showing a preview of the desktop.

C:\Users\LABAdmin\source\repos

Process [20548] C# Console Application.exe - Lifecycle Events - Thread [2124] Main Thread

Program.cs

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

namespace C# Console Application
{
    class Program
    {
        static void Main(string[] args)
        {
            // Creating a Student object
            Student student1 = new Student("Vipul Sunil Patil", 22110189, 88);
            Student student2 = new Student("Ranit Biswas", 22110217, 85);
            Console.WriteLine("==== Student Details ====");
            student1.DisplayDetails();
            Console.WriteLine();
            student2.DisplayDetails();

            Console.WriteLine("\n==== IITGN Student Details ====");
            // Creating a StudentIITGN object
            StudentIITGN iitgnStudent1 = new StudentIITGN("Ranit Biswas", 22110217, 85, "G Hostel");
            iitgnStudent1.DisplayDetails();
        }
    }
}
```

Program.cs

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

namespace C# Console Application
{
    class Student
    {
        // Properties
        public string Name { get; set; }
        public int ID { get; set; }
        public int Marks { get; set; }

        // Constructor
        public Student(string name, int id, int marks)
        {
            Name = name;
            ID = id;
            Marks = marks;
        }

        // Method to calculate grade based on marks
        public string GetGrade()
        {
            if (Marks >= 90)
                return "A+";
            else if (Marks >= 80)
                return "A";
            else if (Marks >= 70)
                return "B+";
            else if (Marks >= 60)
                return "B";
            else if (Marks >= 50)
                return "C+";
            else if (Marks >= 40)
                return "C";
            else
                return "D";
        }
    }
}
```

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

The image displays two side-by-side screenshots of the Microsoft Visual Studio IDE interface, showing the code editor, toolbars, and status bars.

Top Screenshot (Program.cs):

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

namespace CSharp_Console_Application
{
    class Program
    {
        static void Main(string[] args)
        {
            // Creating a Student object
            Student student1 = new Student("Vipul Sunil Patil", 22110189, 88);
            Student student2 = new Student("Ranit Biswas", 22110217, 85);
            Console.WriteLine("==== Student Details ====");
            student1.DisplayDetails();
            Console.WriteLine();
            student2.DisplayDetails();

            Console.WriteLine("\n==== IITGN Student Details ====");
            // Creating a StudentIITGN object
            StudentIITGN iitgnStudent1 = new StudentIITGN("Ranit Biswas", 22110217, 85, "G Hostel");
            iitgnStudent1.DisplayDetails();
        }
    }
}
```

Bottom Screenshot (Program.cs):

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

namespace CSharp_Console_Application
{
    class Program
    {
        static void Main(string[] args)
        {
            // Creating a Student object
            Student student1 = new Student("Vipul Sunil Patil", 22110189, 88);
            Student student2 = new Student("Ranit Biswas", 22110217, 85);
            Console.WriteLine("==== Student Details ====");
            student1.DisplayDetails();
            Console.WriteLine();
            student2.DisplayDetails();

            Console.WriteLine("\n==== IITGN Student Details ====");
            // Creating a StudentIITGN object
            StudentIITGN iitgnStudent1 = new StudentIITGN("Ranit Biswas", 22110217, 85, "G Hostel");
            iitgnStudent1.DisplayDetails();
        }
    }
}
```

The code in both screenshots is identical, demonstrating the creation of `Student` and `StudentIITGN` objects and their `DisplayDetails` method calls. The bottom screenshot shows the code with syntax highlighting and various VS tools like Snipping Tool and Error List visible.

```
== Student Details ==
Name: Vipul Sunil Patil
```

```
base Class: Student
    < Student
        // Properties
        2 references
        public string Name { get; set; } < 1ms elapsed
        2 references
        public int ID { get; set; }
        2 references
        public int Marks { get; set; }

        // Constructor
        3 references
        public Student(string name, int id, int marks)
        {
            Name = name;
            ID = id;
            Marks = marks;
        }

        // Method to calculate grade based on marks
        1 reference
        public string CalculateGrade()
        {
            if (Marks >= 90)
                return "A";
            else if (Marks >= 80)
                return "B";
            else if (Marks >= 70)
                return "C";
            else if (Marks >= 60)
                return "D";
            else
                return "F";
        }
    }
```

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
```

```
base Class: Student
Project: C# Console Application (Ctrl+F2)
Use the dropdown to view and switch to other projects this file may belong to.

class Student
{
    // Properties
    public string Name { get; set; }
    public int ID { get; set; }
    public int Marks { get; set; }

    // Constructor
    public Student(string name, int id, int marks)
    {
        Name = name;
        ID = id;
        Marks = marks;
    }

    // Method to calculate grade based on marks
    public string GetGrade()
    {
        if (Marks >= 90)
            return "A";
        else if (Marks >= 75)
            return "B";
        else if (Marks >= 50)
            return "C";
        else
            return "F";
    }
}
```

== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88

```
Program.cs  C# Console Application
using System;
namespace C# Console Application
{
    class Student
    {
        // Properties
        public string Name { get; set; }
        public int ID { get; set; }
        public int Marks { get; set; }

        // Constructor
        public Student(string name, int id, int marks)
        {
            Name = name;
            ID = id;
            Marks = marks;
        }

        // Method to calculate grade based on marks
        public string GetGrade()
        {
            if (Marks >= 90)
                return "A";
            else if (Marks >= 75)
                return "B";
            else if (Marks >= 50)
                return "C";
            else
                return "F";
        }

        // Method to display student details
        public virtual void DisplayDetails()
        {
        }
    }
}
```

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22118189
Marks: 88
```

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

namespace CSharp_Console_Application
{
    class Program
    {
        static void Main(string[] args)
        {
            // Creating a Student object
            Student student1 = new Student("Vipul Sunil Patil", 22118189, 88);
            Student student2 = new Student("Ranit Biswas", 22118217, 85);
            Console.WriteLine("== Student Details ==");
            student1.DisplayDetails();
            Console.WriteLine();
            student2.DisplayDetails();
            Console.WriteLine("\n== IITGN Student Details ==");
            // Creating a StudentIITGN object
            StudentIITGN iitgnStudent1 = new StudentIITGN("Ranit Biswas", 22118217, 85, "G Hostel");
            iitgnStudent1.DisplayDetails();
        }
    }
}
```

```
class Student
{
    public string Name { get; set; }
    public int ID { get; set; }
    public int Marks { get; set; }

    public Student(string name, int id, int marks)
    {
        Name = name;
        ID = id;
        Marks = marks;
    }

    public void DisplayDetails()
    {
        Console.WriteLine("== Student Details ==");
        Console.WriteLine("Name: " + Name);
        Console.WriteLine("ID: " + ID);
        Console.WriteLine("Marks: " + Marks);
    }
}
```

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B
```

```
== Student Details ==
Name: Ranit Biswas
ID: 22110217
```

```
Program.cs  C# Console Application  Student  Name
using System;
namespace C# Console Application
{
    class Program
    {
        static void Main(string[] args)
        {
            Student s = new Student("Vipul Sunil Patil", 22110189, 88);
            Console.WriteLine(s);
        }
    }

    class Student
    {
        // Properties
        public string Name { get; set; }
        public int ID { get; set; }
        public int Marks { get; set; }

        // Constructor
        public Student(string name, int id, int marks)
        {
            Name = name;
            ID = id;
            Marks = marks;
        }

        // Method to calculate grade based on marks
        public char CalculateGrade()
        {
            if (Marks >= 90)
                return 'A';
            else if (Marks >= 80)
                return 'B';
            else if (Marks >= 70)
                return 'C';
            else if (Marks >= 60)
                return 'D';
            else
                return 'E';
        }
    }
}
```

```
Program.cs  C# Console Application  Marks  Marks
using System;
namespace C# Console Application
{
    class Program
    {
        static void Main(string[] args)
        {
            Student s = new Student("Ranit Biswas", 22110217, 88);
            Console.WriteLine(s);
        }
    }

    class Student
    {
        // Properties
        public string Name { get; set; }
        public int ID { get; set; }
        public int Marks { get; set; }

        // Constructor
        public Student(string name, int id, int marks)
        {
            Name = name;
            ID = id;
            Marks = marks;
        }

        // Method to calculate grade based on marks
        public char CalculateGrade()
        {
            if (Marks >= 90)
                return 'A';
            else if (Marks >= 80)
                return 'B';
            else if (Marks >= 70)
                return 'C';
            else if (Marks >= 60)
                return 'D';
            else
                return 'E';
        }
    }
}
```

== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B
Name: Ranit Biswas
ID: 22110217
Marks: 85

```
public Student(string name, int id, int marks)
{
    Name = name;
    ID = id;
    Marks = marks;
}

// Method to calculate grade based on marks
1 reference
public string GetGrade()
{
    if (Marks >= 90) < 1ms elapsed
        return "A";
    else if (Marks >= 75)
        return "B";
    else if (Marks >= 50)
        return "C";
    else
        return "F";
}

// Method to display student details
5 references
No issues found
```

Autos Locals Watch1 Error List Entire Solution 0 Errors 0 Warnings Search Error List Snipping Tool Screenshot copied to clipboard Automatically saved to screenshots folder. Markup and share

```
using System;
class Class: Student
{
    public class Student
    {
        // Properties
        2 references
        public string Name { get; set; }
        2 references
        public int ID { get; set; }
        5 references
        public int Marks { get; set; } < 1ms elapsed

        // Constructor
        1 reference
        public Student(string name, int id, int marks)
        {
            Name = name;
            ID = id;
            Marks = marks;
        }

        // Method to calculate grade based on marks
        1 reference
        No issues found
```

Autos Locals Watch1 Error List Entire Solution 0 Errors 0 Warnings Search Error List Snipping Tool Screenshot copied to clipboard Automatically saved to screenshots folder. Markup and share

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
```

```
using System;
class Program
{
    static class Student
    {
        public string Name { get; set; }
        public int ID { get; set; }
        public int Marks { get; set; }
    }

    // Method to calculate grade based on marks
    public string GetGrade()
    {
        if (Marks >= 90)
            return "A";
        else if (Marks >= 75)
            return "B";
        else if (Marks >= 50)
            return "C";
        else
            return "F";
    }

    // Method to display student details
    public virtual void DisplayDetails()
    {
    }
}
```

```
using System;
class Program
{
    class Student
    {
        // Properties
        public string Name { get; set; }
        public int ID { get; set; }
        public int Marks { get; set; }
    }

    // Constructor
    public Student(string name, int id, int marks)
    {
        Name = name;
        ID = id;
        Marks = marks;
    }

    // Method to calculate grade based on marks
    public string GetGrade()
    {
    }

    // Method to display student details
    public void DisplayDetails()
    {
    }
}
```

```
38°C ENG < > 10:44 AM

C:\Users\LABAdmin\source\repos + x Search Help
Process: [20548] CS Console Application.exe | Lifecycle Events | Thread: [2124] Main Thread
Programs x
CS Console Application - % Student - Student(string name, int id, int marks)
References
is Student
Properties
    public string Name { get; set; }
    public int ID { get; set; }
    public int Marks { get; set; }

Constructor
public Student(string name, int id, int marks) { Name = name; ID = id; Marks = marks; }

Method to calculate grade based on marks
public string GetGrade()
{
    if (Marks >= 90)
        return "A";
}

110% Autos Search (Ctrl+F) 3 x Error List
Name Value Type
Entire Solution 0 Errors 0 Warnings
Search Error List
Code Snipping Tool ...
Screenshot copied to clipboard
Automatically saved to screenshots folder.

Screenshot copied to clipboard
Markup and share

38°C EN F9 dx 10:44 AM
```

38°C ENG < > 10:44 AM

C:\Users\LABAdmin\source\repos + x Search Help

Process: [20548] CS Console Application.exe | Lifecycle Events | Thread: [2124] Main Thread

Programs x

CS Console Application - % Student - Student(string name, int id, int marks)

References

is Student

Properties

 public string Name { get; set; }

 public int ID { get; set; }

 public int Marks { get; set; }

Constructor

public Student(string name, int id, int marks) { Name = name; ID = id; Marks = marks; }

Method to calculate grade based on marks

public string GetGrade()

{

 if (Marks >= 90)

 return "A";

}

110% Autos Search (Ctrl+F) 3 x Error List

Name Value Type

Entire Solution 0 Errors 0 Warnings

Search Error List

Code Snipping Tool ...

Screenshot copied to clipboard

Automatically saved to screenshots folder.

Markup and share

38°C EN F9 dx 10:44 AM

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
```

```
// Creating a Student object
Student student1 = new Student("Vipul Sunil Patil", 22110189, 88);
Student student2 = new Student("Ranit Biswas", 22110217, 85);
Console.WriteLine("== Student Details ==");
student1.DisplayDetails();
Console.WriteLine();
student2.DisplayDetails();

Console.WriteLine("== IITGN Student Details ==");
StudentIITGN iitgnStudent1 = new StudentIITGN("Ranit Biswas", 22110217, 85, "G Hostel");
iitgnStudent1.DisplayDetails();
```

```
Java Class: Student
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
```

```
class Student
{
    // Properties
    public string Name { get; set; }
    public int ID { get; set; }
    public int Marks { get; set; }

    // Constructor
    public Student(string name, int id, int marks)
    {
        Name = name;
        ID = id;
        Marks = marks;
    }

    // Method to calculate grade based on marks
    public char CalculateGrade()
    {
        if (Marks >= 90)
            return 'A';
        else if (Marks >= 80)
            return 'B';
        else if (Marks >= 70)
            return 'C';
        else if (Marks >= 60)
            return 'D';
        else
            return 'E';
    }
}
```

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
Name: Ranit Biswas
```

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Student s1 = new Student("Vipul Sunil Patil", 22110189, 88);
            Student s2 = new Student("Ranit Biswas", 22110217, 85);

            Console.WriteLine("== Student Details ==");
            Console.WriteLine("Name: " + s1.Name);
            Console.WriteLine("ID: " + s1.ID);
            Console.WriteLine("Marks: " + s1.Marks);
            Console.WriteLine("Grade: " + s1.Grade);

            Console.WriteLine("\n== IITGN Student Details ==");
            Console.WriteLine("Name: " + s2.Name);
            Console.WriteLine("ID: " + s2.ID);
            Console.WriteLine("Marks: " + s2.Marks);
            Console.WriteLine("Grade: " + s2.Grade);
        }
    }

    class Student
    {
        public string Name { get; set; }
        public int ID { get; set; }
        public int Marks { get; set; }

        public Student(string name, int id, int marks)
        {
            Name = name;
            ID = id;
            Marks = marks;
        }

        public char CalculateGrade()
        {
            if (Marks >= 90)
                return 'A';
            else if (Marks >= 80)
                return 'B';
            else if (Marks >= 70)
                return 'C';
            else if (Marks >= 60)
                return 'D';
            else
                return 'E';
        }
    }
}
```

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
Name: Ranit Biswas
ID: 22110217
Marks: 85
```

This screenshot shows the Visual Studio IDE interface. The code editor displays a C# file named 'Program.cs' containing the original 'Student' class definition. The class has properties for Name, ID, and Marks, and a method 'GetGrade()' that returns a grade based on the marks. The 'Solution Explorer' and 'Error List' panes are visible on the right.

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
Name: Ranit Biswas
ID: 22110217
Marks: 85
```

This screenshot shows the Visual Studio IDE interface again. The code editor now displays a refactored version of the 'Program.cs' file. The 'Student' class is now derived from a base class named 'Marks'. The 'Marks' class contains the properties and methods of the original 'Student' class. The 'Student' class itself is now empty except for its constructor and the inherited 'GetGrade()' method. The 'Solution Explorer' and 'Error List' panes are visible on the right.

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
Name: Ranit Biswas
ID: 22110217
Marks: 85
```

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

namespace CSharp_Console_Application
{
    class Program
    {
        static void Main(string[] args)
        {
            // Method to calculate grade based on marks
            public string GetGrade()
            {
                if (Marks >= 90)
                    return "A";
                else if (Marks >= 75)
                    return "B";
                else if (Marks >= 50)
                    return "C";
                else
                    return "F";
            }

            // Method to display student details
            public virtual void DisplayDetails()
            {
            }
        }
    }
}
```

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B
```

```
public StudentIITGN(string name, int id, int marks, string hostelName)
    : base(name, id, marks)
{
    Hostel_Name_IITGN = hostelName;
}

// Override DisplayDetails method to include hostel info
public override void DisplayDetails()
{
    base.DisplayDetails();
    Console.WriteLine($"Hostel Name (IITGN): {Hostel_Name_IITGN}");
}

class Program
{
    static void Main(string[] args)
    {
        // Creating a Student object
        Student student1 = new Student("Vipul Sunil Patil", 22110189, 88);
        Student student2 = new Student("Ranit Biswas", 22110217, 85);
    }
}
```

```
== Student Details ==
Name: Vipul Sunil Patil
ID: 22110189
Marks: 88
Grade: B

Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B

== IITGN Student Details ==
Name: Ranit Biswas
ID: 22110217
Marks: 85
Grade: B
Hostel Name (IITGN): G Hostel

C:\Users\LABAdmin\source\repos\CH Console Application\CH Console Application\bin\Debug\net8.0\CH Console Application.exe (process 20548) 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 . . .
```

```
Output
Show output from: Debug
The thread '.NET IP Worker' (29732) has exited with code 0 (0x0).
The thread '.NET IP Worker' (30002) has exited with code 0 (0x0).
'CH Console Application.exe' (CoreCLR: clhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NET\8.0.0\Microsoft.CSharp.dll'.
'CH Console Application.exe' (CoreCLR: clhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NET\8.0.0\System.Numerics.Vectors.dll'.
The program '[20548] CH Console Application.exe' has exited with code 0 (0x0).
```

For Activities (4)

The image displays two side-by-side screenshots of the Microsoft Visual Studio IDE interface, likely from different sessions or configurations.

Top Screenshot:

- Code Editor:** Shows the `Program.cs` file of a C# Console Application project. The code defines a `Program` class with a `Main` method. It creates a `Calculator` instance, calls `GetInput()`, and `PerformOperations()`. A `try-catch` block handles exceptions, and a `Console.WriteLine` statement prints an error message if an exception occurs.
- Error List:** Shows two warnings (CS8604) related to possible null reference arguments for parameters 's' in `int.Parse(string s)`.

Bottom Screenshot:

- Code Editor:** Shows the `Calculator.cs` file of the same project. The `Calculator` class has a `GetInput` method that reads user input from the console. It uses a `try-catch` block to handle `FormatException` and prints an error message if the input is invalid.
- Error List:** Shows two errors (CS860) related to possible null reference arguments for parameters 's' in `int.Parse(string s)`.

Both screenshots show the Windows taskbar at the bottom with various pinned icons like File Explorer, Edge, and File History.

The screenshot displays two instances of the Microsoft Visual Studio IDE, both showing a C# console application project named "Calculator".

Code Editor:

```
using System;
class Calculator
{
    private int number1;
    private int number2;

    public void GetInput()
    {
        Console.Write("Enter the first number: ");
        while (true)
        {
            try
            {
                if (s < 2ms elapsed)
                    number1 = int.Parse(Console.ReadLine());
                break; // Exit loop if input is valid
            }
            catch (FormatException)
            {
                Console.WriteLine("Error: Invalid input! Please enter an integer.");
                Console.Write("Enter the first number: ");
            }
        }
    }

    public void AddNumbers()
    {
        Console.Write("Enter the second number: ");
        while (true)
        {
            try
            {
                number2 = int.Parse(Console.ReadLine());
                break; // Exit loop if input is valid
            }
            catch (FormatException)
            {
                Console.WriteLine("Error: Invalid input! Please enter an integer.");
                Console.Write("Enter the second number: ");
            }
        }
    }

    public void ShowResult()
    {
        Console.WriteLine($"The sum of {number1} and {number2} is {number1 + number2}.");
    }
}
```

Error List:

Code	Description	Project	File
CS8604	Possible null reference argument for parameter 's' in 'int.Parse(string s)'.	C# Console Application	Program.cs
CS8604	Possible null reference argument for parameter 's' in 'int.Parse(string s)'.	C# Console Application	Program.cs

Output Window:

```
Enter the first number: 6
Enter the second number: 7
The sum of 6 and 7 is 13.
```

Enter the first number: 6
Enter the second number: 5

```
Program.cs  C# Console Application  Calculator  PerformOperations()

    catch (FormatException)
    {
        Console.WriteLine("Error: Invalid input! Please enter an integer.");
        Console.Write("Enter the second number: ");
    }

    1 reference
    public void PerformOperations()
    {
        int sum = number1 + number2; < 1ms elapsed
        int difference = number1 - number2;
        int product = number1 * number2;

        try
        {
            // Handle division by zero using try-catch
            double division = (double)number1 / number2;
            Console.WriteLine($"Division: {division}");
        }
        catch (DivideByZeroException)
        {
    }
```

Autos Locals Watch1 Error List

Name	Value	Type
number1	6	int
number2	5	int
sum	0	int
this	(Calculator)	Calculator

Entire Solution 0 Errors 2 Warnings

Code	Description	Project	File
CS8604	Possible null reference argument for parameter 's' in 'int.Parse(string s)'.	C# Console Application	Program.cs
CS8604	Possible null reference argument for parameter 's' in 'int.Parse(string s)'.	C# Console Application	Program.cs

Enter the first number: 6
Enter the second number: 5

```
Program.cs  C# Console Application  Calculator  PerformOperations()

    catch (FormatException)
    {
        Console.WriteLine("Error: Invalid input! Please enter an integer.");
        Console.Write("Enter the second number: ");
    }

    1 reference
    public void PerformOperations()
    {
        int sum = number1 + number2; < 1ms elapsed
        int difference = number1 - number2; < 1ms elapsed
        int product = number1 * number2;

        try
        {
            // Handle division by zero using try-catch
            double division = (double)number1 / number2;
            Console.WriteLine($"Division: {division}");
        }
        catch (DivideByZeroException)
        {
    }
```

Autos Locals Watch1 Error List

Name	Value	Type
difference	0	int
number1	6	int
number2	5	int
sum	11	int
this	(Calculator)	Calculator

Entire Solution 0 Errors 2 Warnings

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

```
Enter the first number: 6
Enter the second number: 5

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

namespace CSharp_Console_Application
{
    class Program
    {
        static void Main(string[] args)
        {
            Calculator calculator = new Calculator();
            calculator.PerformOperations();
        }
    }

    class Calculator
    {
        public void PerformOperations()
        {
            int number1 = 6;
            int number2 = 5;
            int sum = number1 + number2;
            int difference = number1 - number2;
            int product = number1 * number2; // 30ms elapsed

            try
            {
                // Handle division by zero using try-catch
                double division = (double)number1 / number2;
                Console.WriteLine($"Division: {division}");
            }
            catch (DivideByZeroException)
            {
                Console.WriteLine("Error: Division by zero!");
            }
        }
    }
}
```

Autos

Name	Value	Type
difference	1	int
number1	6	int
number2	5	int
product	0	int
this	(Calculator)	Calculator

Watch 1

Error List

Entire Solution 0 Errors 2 Warnings

CS860 CS860

Snipping Tool Screenshot copied to clipboard Automatically saved to screenshots folder.

Markup and share

```
Enter the first number: 6
Enter the second number: 5

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

namespace CSharp_Console_Application
{
    class Program
    {
        static void Main(string[] args)
        {
            Calculator calculator = new Calculator();
            calculator.PerformOperations();
        }
    }

    class Calculator
    {
        public void PerformOperations()
        {
            int number1 = 6;
            int number2 = 5;
            int sum = number1 + number2;
            int difference = number1 - number2;
            int product = number1 * number2; // 30ms elapsed

            try
            {
                // Handle division by zero using try-catch
                double division = (double)number1 / number2;
                Console.WriteLine($"Division: {division}");
            }
            catch (DivideByZeroException)
            {
                Console.WriteLine("Error: Division by zero!");
            }
        }
    }
}
```

Autos

Name	Value	Type
difference	1	int
number1	6	int
number2	5	int
product	0	int
this	(Calculator)	Calculator

Watch 1

Error List

Entire Solution 0 Errors 2 Warnings

CS860 CS860

Snipping Tool Screenshot copied to clipboard Automatically saved to screenshots folder.

Markup and share

```

C:\Users\LABAdmin\source\repos\C# Console Application\C# Console Application\bin\Debug\net8.0\C# Console Application.exe (process 11900) 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 . . .

```

```

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

namespace C# Console Application
{
    class Calculator
    {
        public void PerformOperations()
        {
            int sum = number1 + number2;
            int difference = number1 - number2;
            int product = number1 * number2;

            try
            {
                // Handle division by zero using try-catch
                double division = (double)number1 / number2;
                Console.WriteLine($"Division: {division}");
            }
            catch (DivideByZeroException)
            {
                Console.WriteLine("Error: Division by zero is not allowed.");
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Unexpected error: {ex.Message}");
            }

            Console.WriteLine($"Sum: {sum}");
            Console.WriteLine($"Difference: {difference}");
            Console.WriteLine($"Product: {product}");

            // Check if the sum is even or odd
            if (sum % 2 == 0)
                Console.WriteLine("The sum is even.");
            else
                Console.WriteLine("The sum is odd.");
        }
    }
}

Enter the first number: 6
Enter the second number: 5
Division: 1.2
Sum: 11
Difference: 1
Product: 30
The sum is odd.

Process finished with exit code 0 (0x0)

```

APPENDIX

I would like to express my sincere gratitude to my course instructor, Prof. [Shouwick Mondal](#), for his invaluable guidance and support throughout this lab. I also appreciate the assistance from All TAs, whose help with me with troubleshooting.

Additionally, I am grateful for the resources provided, including Which I have sited down which helped me resolve issues efficiently. Finally, I'd like to thank my peers for contributing to a collaborative and supportive learning environment.

CITATION

- [1] BillWagner. (n.d.-a). *C# Guide - .NET managed language*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/csharp/>
- [2] BillWagner. (n.d.-b). *Exception Handling - C#*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/exceptions/exception-handling>
- [3] BillWagner. (n.d.-c). *Exceptions and Exception Handling - C#*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/exceptions/>
- [4] BillWagner. (n.d.-d). *Tutorial: Introduction to Inheritance - C#*. Microsoft Learn. <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/tutorials/inheritance>
- [5] CS202_Lecture_9.pdf. (n.d.). Google Docs. <https://drive.google.com/file/d/1bJG8oi3f2s-XBT3OZEEZ0idv-B4IzSAA/view>
- [6] CS202_Lecture_10.pdf. (n.d.). Google Docs. <https://drive.google.com/file/d/12-KtkebmYLFGosvB3tyHNEVQYe--tkbV/view>
- [7] Dotnet. (n.d.). *GitHub - dotnet/dotnet-console-games: Game examples implemented as .NET console applications primarily for providing education and inspiration.* GitHub. <https://github.com/dotnet/dotnet-console-games>
- [8] Download .NET (Linux, MacOS, and Windows). (n.d.). Microsoft. <https://dotnet.microsoft.com/en-us/download>
- [9] Mikejo. (n.d.). *Debugger documentation - Visual Studio (Windows)*. Microsoft Learn. <https://learn.microsoft.com/en-us/visualstudio/debugger/?view=vs-2022>
- [10] *Visual Studio: IDE and code editor for software developers and teams*. (2025, April 8). Visual Studio. <https://visualstudio.microsoft.com/>