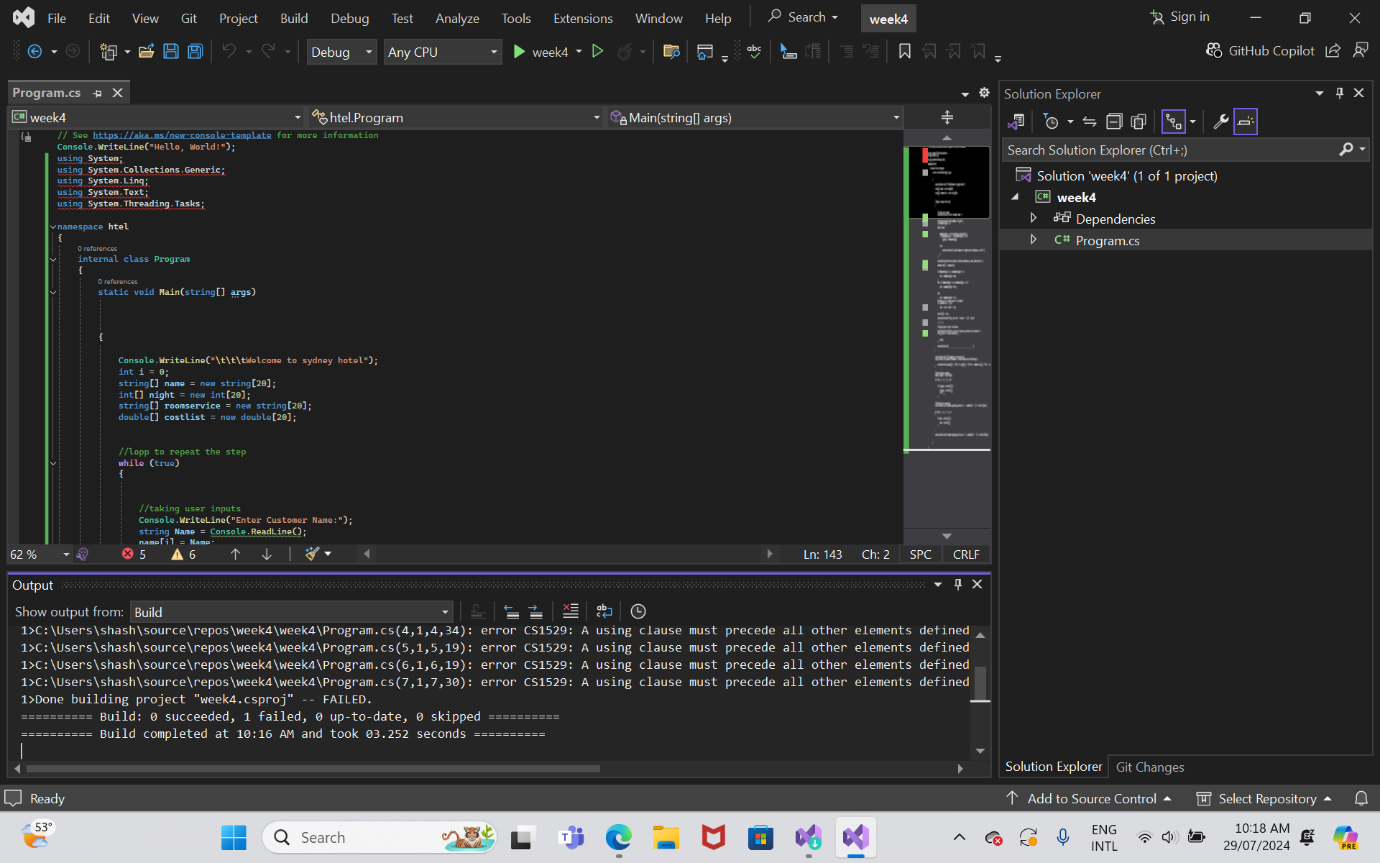
Week 4 Activity

Task 1: Download and explore the code using Visual Studio 2019 or 2022. Add a screenshot of the code from Visual Studio and include it in the report.



Task 2: Describe the problem that the coder is solving in the program.

1. Refactor the code: Improve readability, maintainability, and performance.
2. Add error handling: Make the code more robust by handling potential errors more gracefully.
3. Add new features: Implement additional functionality or features.
4. Optimize the code: Identify and improve any inefficient parts of the code.
5. Code comments and documentation: Add comments and documentation for better understanding.

Task 3: Consider redesigning and recoding the program by identifying four changes in the C# code using functions, classes, lists, etc. For each change, you need to display a code snippet screenshot before the change and another after the changes, accompanied by a description of what you modified.

Change 1: Extract Reservation Logic into a Separate Class

Before:

The reservation logic is embedded directly within the Main method.

The reservation logic is embedded directly within the Main method.

csharp

Copy code

// Part of Main method handling reservations

string[] name = new string[20];

int[] night = new int[20];

string[] roomservice = new string[20];

double[] costlist = new double[20];

After:

Create a Reservation class to encapsulate reservation details and logic.

A computer screen shot of a black screen

Description automatically generated

**Change 2: Use Lists Instead of Arrays**

**Before:**

Fixed-size arrays are used, which limits the flexibility.

csharp

Copy code

string[] name = new string[20];

int[] night = new int[20];

string[] roomservice = new string[20];

double[] costlist = new double[20];

After:

Use List<T> for dynamic storage of reservations.

List<Reservation> reservations = new List<Reservation>();

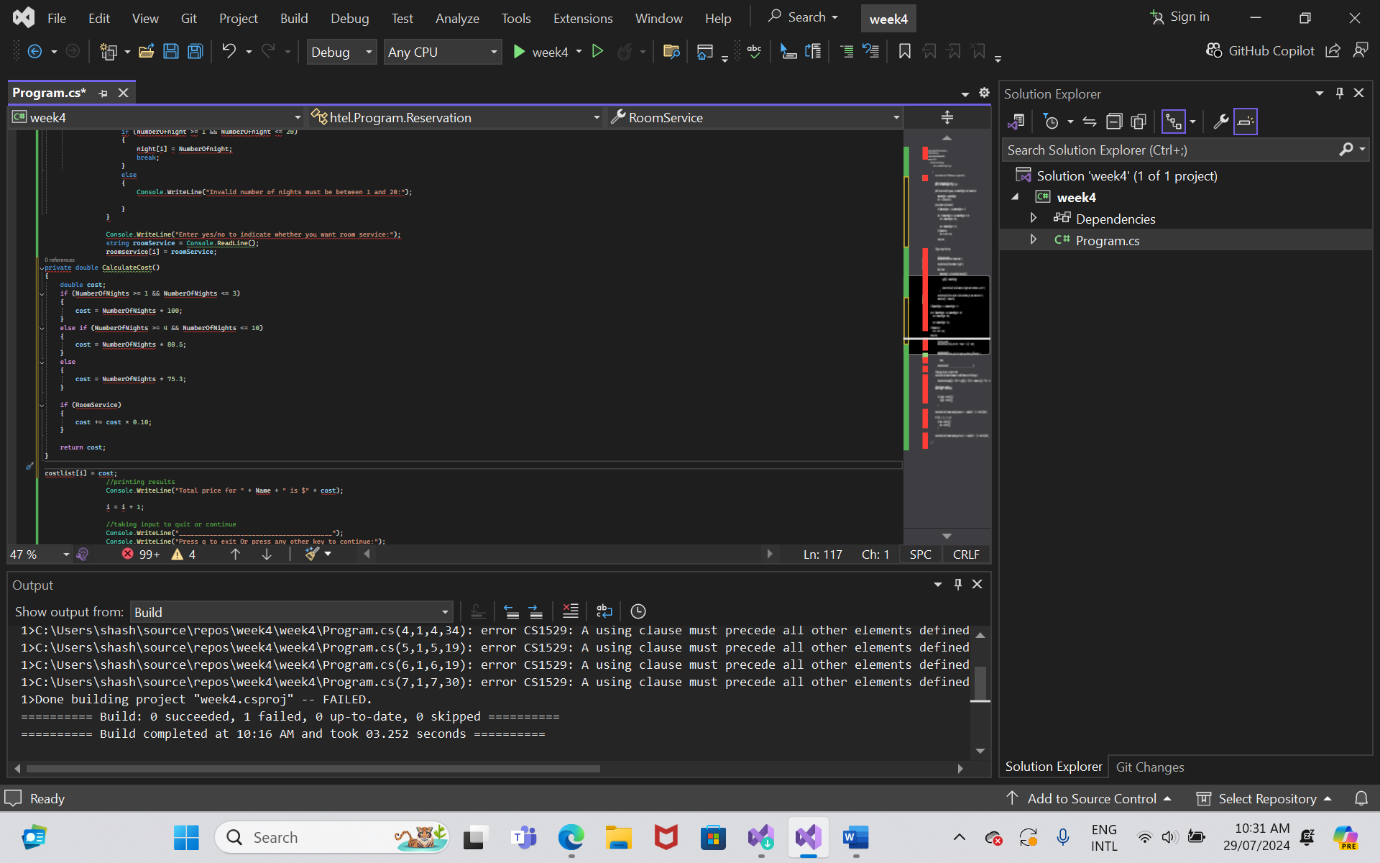
**Change 3: Encapsulate Cost Calculation Logic into a Method**

**Before:**

Cost calculation logic is repeated within the Main method.

**After:**

Encapsulate cost calculation within the Reservation class.



**Change 4: Improve User Interaction and Input Validation**

**Before:**

User input and validation logic are intermixed, making the code less readable.

**After:**

Encapsulate user input and validation into separate methods for clarity.

private static int GetNumberOfNights()

{

int numberOfNights;

while (true)

{

Console.WriteLine("Enter Number of nights:");

if (int.TryParse(Console.ReadLine(), out numberOfNights) && numberOfNights >= 1 && numberOfNights <= 20)

{

break;

}

else

{

Console.WriteLine("Invalid number of nights. Must be between 1 and 20.");

}

}

return numberOfNights;

}