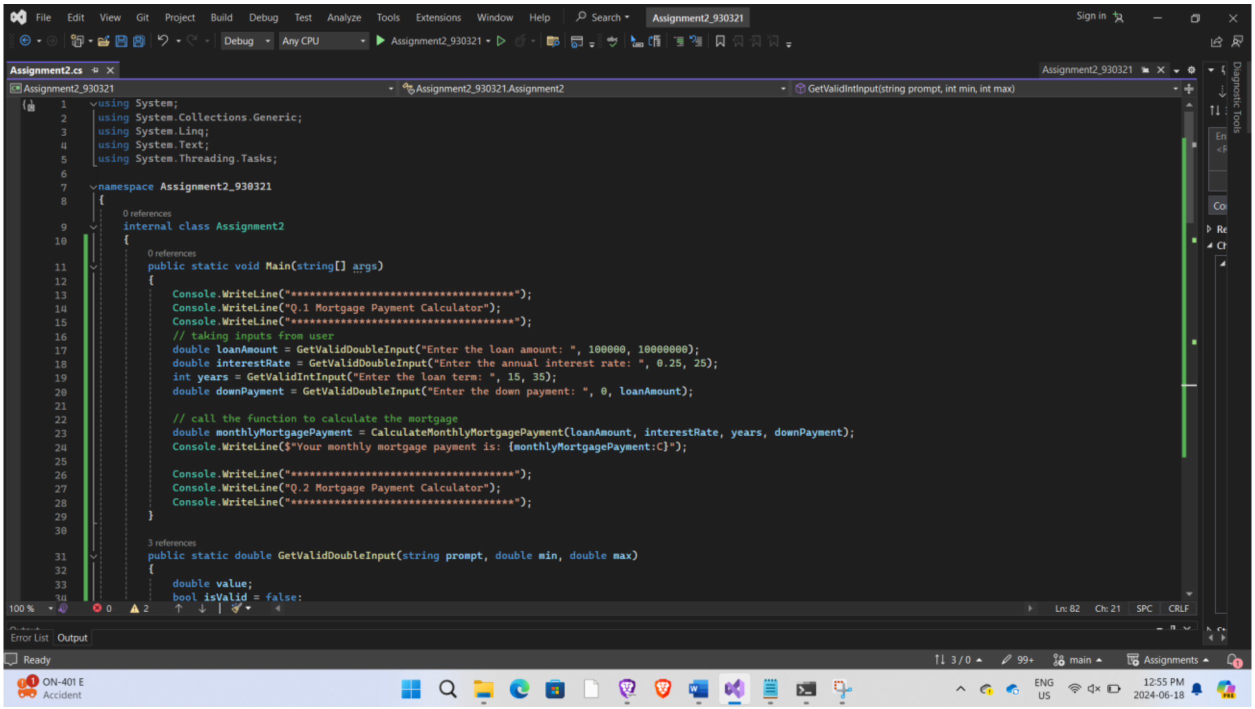
CSharp

Student Name : Shreejana Shrestha

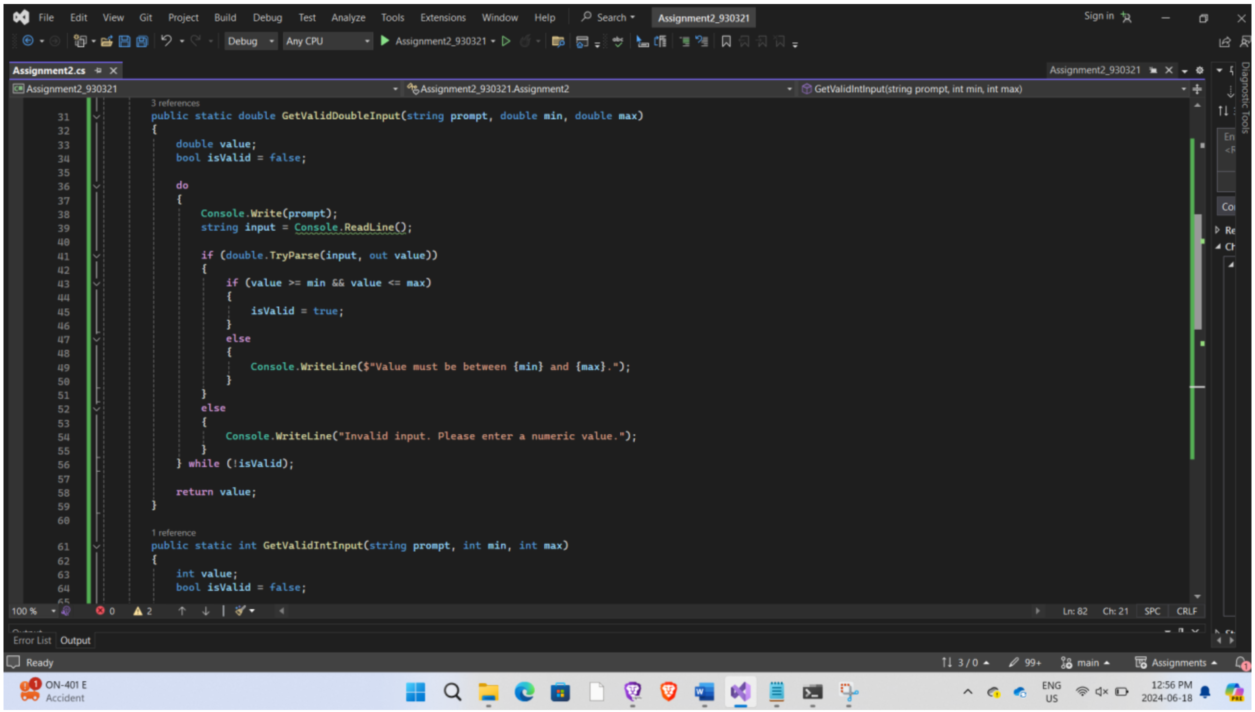
Student Id : C0930321

# Question 1

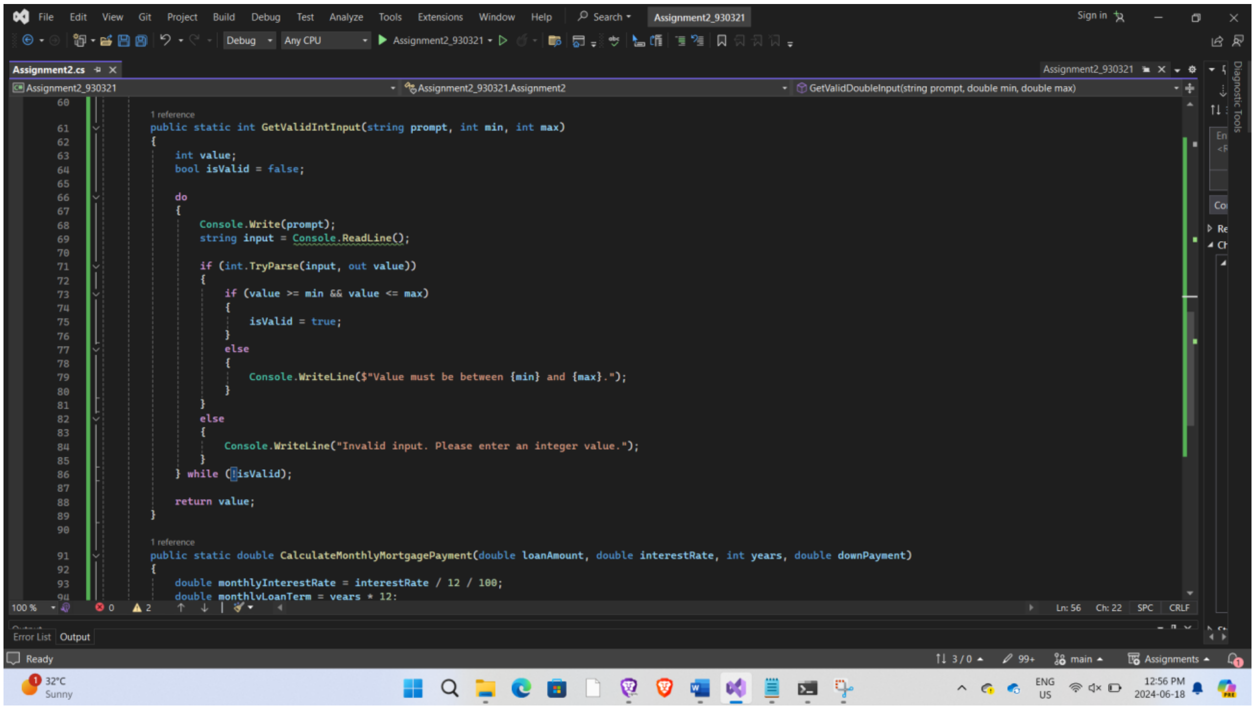
# Code screenshot



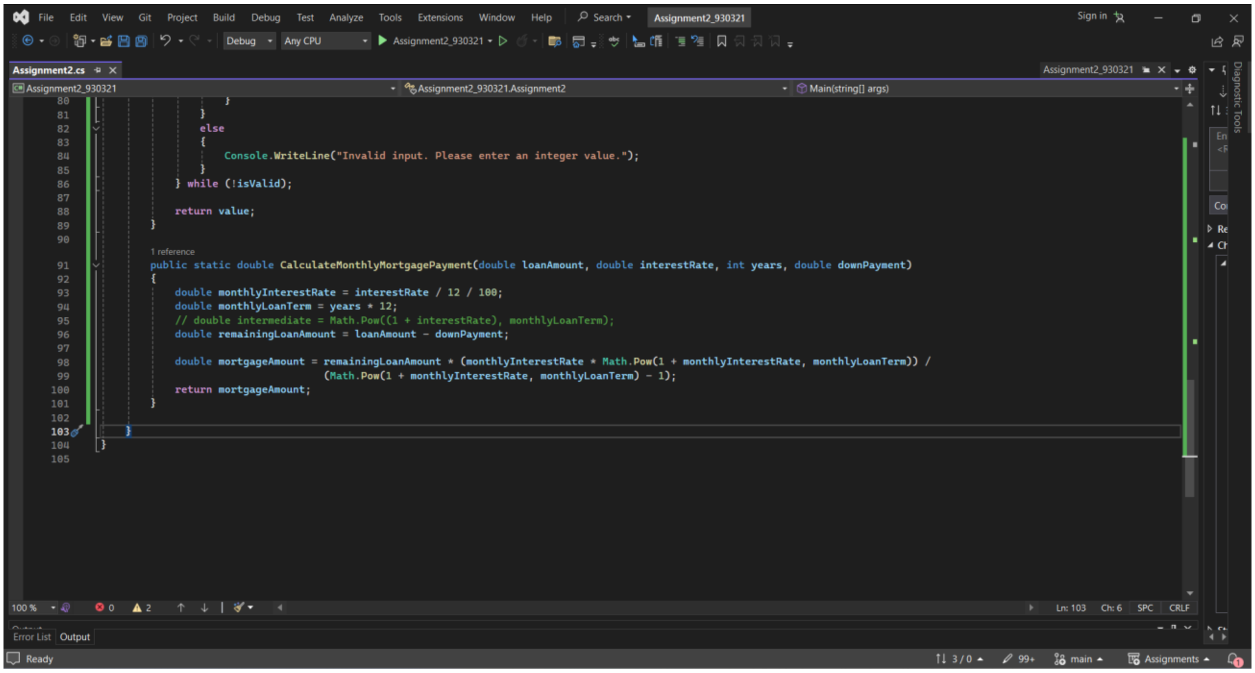
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Assignment2_930321
{
    internal class Assignment2
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("************************************");
            Console.WriteLine("Q.1 Mortgage Payment Calculator");
            Console.WriteLine("************************************");
            // taking inputs from user
            double loanAmount = GetValidDoubleInput("Enter the loan amount: ", 100000, 10000000);
            double interestRate = GetValidDoubleInput("Enter the annual interest rate: ", 0.25, 25);
            int years = GetValidIntInput("Enter the loan term: ", 15, 35);
            double downPayment = GetValidDoubleInput("Enter the down payment: ", 0, loanAmount);

            // call the function to calculate the mortgage
            double monthlyMortgagePayment = CalculateMonthlyMortgagePayment(loanAmount, interestRate, years, downPayment);
            Console.WriteLine($"Your monthly mortgage payment is: {monthlyMortgagePayment:C}");

            Console.WriteLine("************************************");
            Console.WriteLine("Q.2 Mortgage Payment Calculator");
            Console.WriteLine("************************************");
        }

        public static double GetValidDoubleInput(string prompt, double min, double max)
        {
            double value;
            bool isValid = false;
```



```csharp
public static double GetValidDoubleInput(string prompt, double min, double max)
{
    double value;
    bool isValid = false;

    do
    {
        Console.Write(prompt);
        string input = Console.ReadLine();

        if (double.TryParse(input, out value))
        {
            if (value >= min && value <= max)
            {
                isValid = true;
            }
            else
            {
                Console.WriteLine($"Value must be between {min} and {max}.");
            }
        }
        else
        {
            Console.WriteLine("Invalid input. Please enter a numeric value.");
        }
    } while (!isValid);

    return value;
}

public static int GetValidIntInput(string prompt, int min, int max)
{
    int value;
    bool isValid = false;
```



```csharp
public static int GetValidIntInput(string prompt, int min, int max)
{
    int value;
    bool isValid = false;

    do
    {
        Console.Write(prompt);
        string input = Console.ReadLine();

        if (int.TryParse(input, out value))
        {
            if (value >= min && value <= max)
            {
                isValid = true;
            }
            else
            {
                Console.WriteLine($"Value must be between {min} and {max}.");
            }
        }
        else
        {
            Console.WriteLine("Invalid input. Please enter an integer value.");
        }
    } while (!isValid);

    return value;
}

public static double CalculateMonthlyMortgagePayment(double loanAmount, double interestRate, int years, double downPayment)
{
    double monthlyInterestRate = interestRate / 12 / 100;
    double monthlyLoanTerm = years * 12;
```



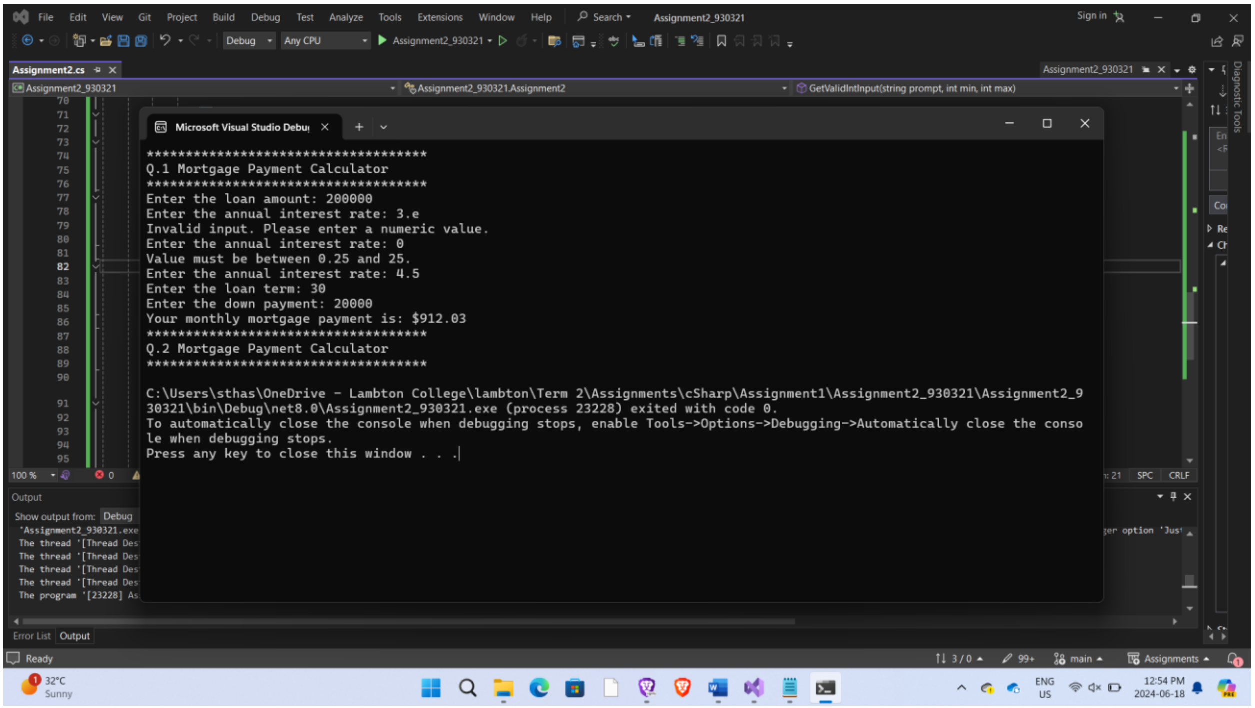
```
                    }
                }
                else
                {
                    Console.WriteLine("Invalid input. Please enter an integer value.");
                }
            } while (!isValid);

            return value;
        }

        public static double CalculateMonthlyMortgagePayment(double loanAmount, double interestRate, int years, double downPayment)
        {
            double monthlyInterestRate = interestRate / 12 / 100;
            double monthlyLoanTerm = years * 12;
            // double intermediate = Math.Pow((1 + interestRate), monthlyLoanTerm);
            double remainingLoanAmount = loanAmount - downPayment;

            double mortgageAmount = remainingLoanAmount * (monthlyInterestRate * Math.Pow(1 + monthlyInterestRate, monthlyLoanTerm)) /
                                    (Math.Pow(1 + monthlyInterestRate, monthlyLoanTerm) - 1);
            return mortgageAmount;
        }

    }
}
```

# code

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

namespace Assignment2_930321
{
    internal class Assignment2
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("*************************************");
            Console.WriteLine("Q.1 Mortgage Payment Calculator");
            Console.WriteLine("*************************************");
            // taking inputs from user
            double loanAmount = GetValidDoubleInput("Enter the loan amount: ", 
100000, 10000000);
            double interestRate = GetValidDoubleInput("Enter the annual interest 
rate: ", 0.25, 25);
            int years = GetValidIntInput("Enter the loan term: ", 15, 35);
            double downPayment = GetValidDoubleInput("Enter the down payment: ", 0, 
loanAmount);

            // call the function to calculate the mortgage
            double monthlyMortgagePayment = 
CalculateMonthlyMortgagePayment(loanAmount, interestRate, years, downPayment);
            Console.WriteLine($"Your monthly mortgage payment is: 
{monthlyMortgagePayment:C}");
```

```
            Console.WriteLine("************************************");
            Console.WriteLine("Q.2 ");
            Console.WriteLine("************************************");

        }

        public static double GetValidDoubleInput(string prompt, double min, double
max)
        {
            double value;
            bool isValid = false;

            do
            {
                Console.Write(prompt);
                string input = Console.ReadLine();

                if (double.TryParse(input, out value))
                {
                    if (value >= min && value <= max)
                    {
                        isValid = true;
                    }
                    else
                    {
                        Console.WriteLine($"Value must be between {min} and
{max}.");
                    }
                }
                else
                {
                    Console.WriteLine("Invalid input. Please enter a numeric
value.");
                }
            } while (!isValid);

            return value;
        }

        public static int GetValidIntInput(string prompt, int min, int max)
        {
            int value;
            bool isValid = false;

            do
            {
                Console.Write(prompt);
                string input = Console.ReadLine();

                if (int.TryParse(input, out value))
                {
                    if (value >= min && value <= max)
                    {
                        isValid = true;
                    }
                    else
                    {
                        Console.WriteLine($"Value must be between {min} and 
{max}.");
                    }
                }
                else
                {
                    Console.WriteLine("Invalid input. Please enter an integer 
value.");
                }
            } while (!isValid);

            return value;
        }

        public static double CalculateMonthlyMortgagePayment(double loanAmount, 
double interestRate, int years, double downPayment)
        {
            double monthlyInterestRate = interestRate / 12 / 100;
            double monthlyLoanTerm = years * 12;
            // double intermediate = Math.Pow((1 + interestRate), monthlyLoanTerm);
            double remainingLoanAmount = loanAmount - downPayment;

            double mortgageAmount = remainingLoanAmount * (monthlyInterestRate * 
Math.Pow(1 + monthlyInterestRate, monthlyLoanTerm)) /
                                    (Math.Pow(1 + monthlyInterestRate, 
monthlyLoanTerm) - 1);
            return mortgageAmount;
        }

    }
}
```

# output Screenshot

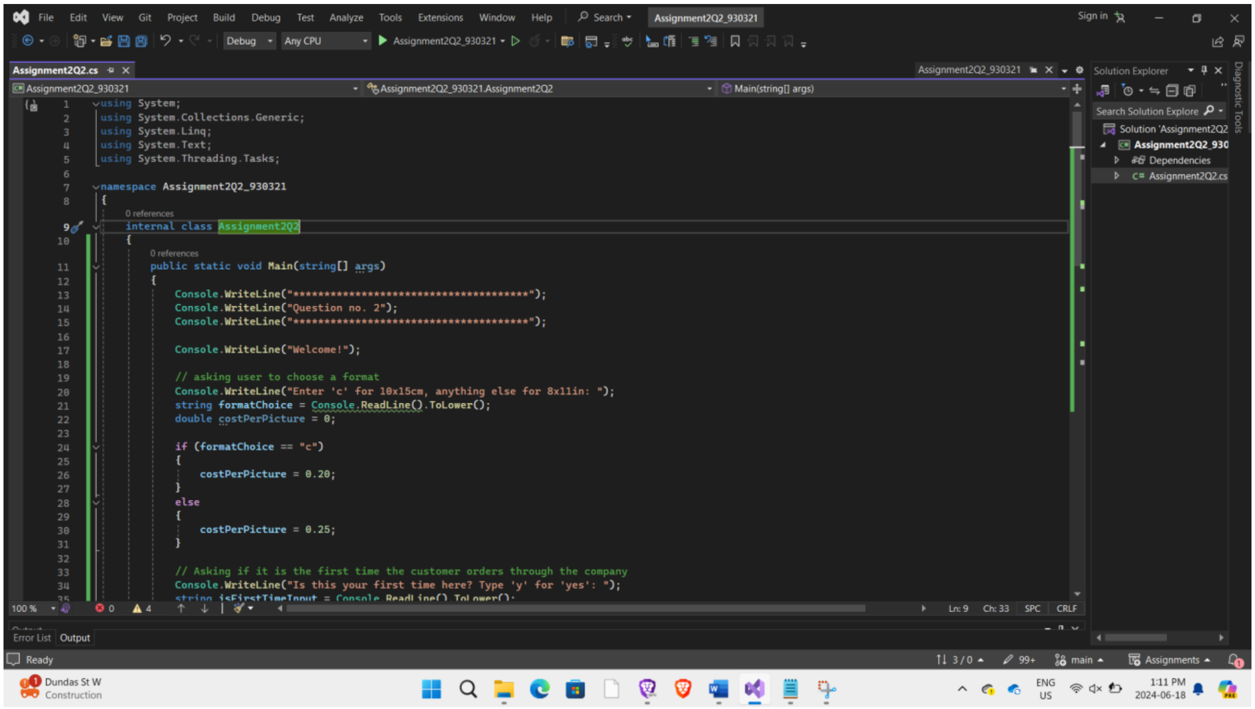


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
Assignment2_930321
Sign in
Debug
Any CPU
Assignment2_930321
Assignment2.cs
Assignment2_930321.Assignment2
GetValidIntInput(string prompt, int min, int max)
Microsoft Visual Studio Debu
Output
Show output from: Debug
Error List
Ready
main
Assignments
32°C
Sunny
ENG
US
12:54 PM
2024-06-18

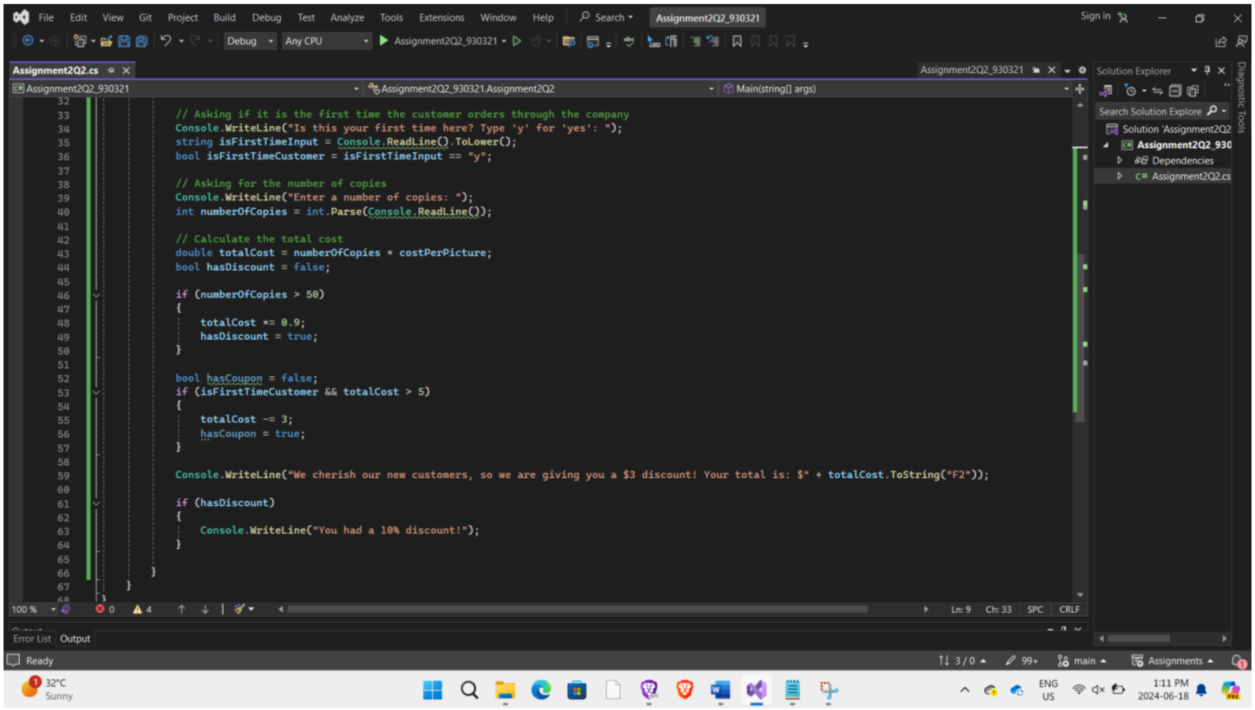
```
************************************
Q.1 Mortgage Payment Calculator
************************************
Enter the loan amount: 200000
Enter the annual interest rate: 3.e
Invalid input. Please enter a numeric value.
Enter the annual interest rate: 0
Value must be between 0.25 and 25.
Enter the annual interest rate: 4.5
Enter the loan term: 30
Enter the down payment: 20000
Your monthly mortgage payment is: $912.03
************************************
Q.2 Mortgage Payment Calculator
************************************

C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\cSharp\Assignment1\Assignment2_930321\Assignment2_9
30321\bin\Debug\net8.0\Assignment2_930321.exe (process 23228) exited with code 0.
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 . . .
```

# Question 2

# Code screenshot



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

namespace Assignment2Q2_930321
{
    internal class Assignment2Q2
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("**************************************");
            Console.WriteLine("Question no. 2");
            Console.WriteLine("**************************************");

            Console.WriteLine("Welcome!");

            // asking user to choose a format
            Console.WriteLine("Enter 'c' for 10x15cm, anything else for 8x11in: ");
            string formatChoice = Console.ReadLine().ToLower();
            double costPerPicture = 0;

            if (formatChoice == "c")
            {
                costPerPicture = 0.20;
            }
            else
            {
                costPerPicture = 0.25;
            }

            // Asking if it is the first time the customer orders through the company
            Console.WriteLine("Is this your first time here? Type 'y' for 'yes': ");
```

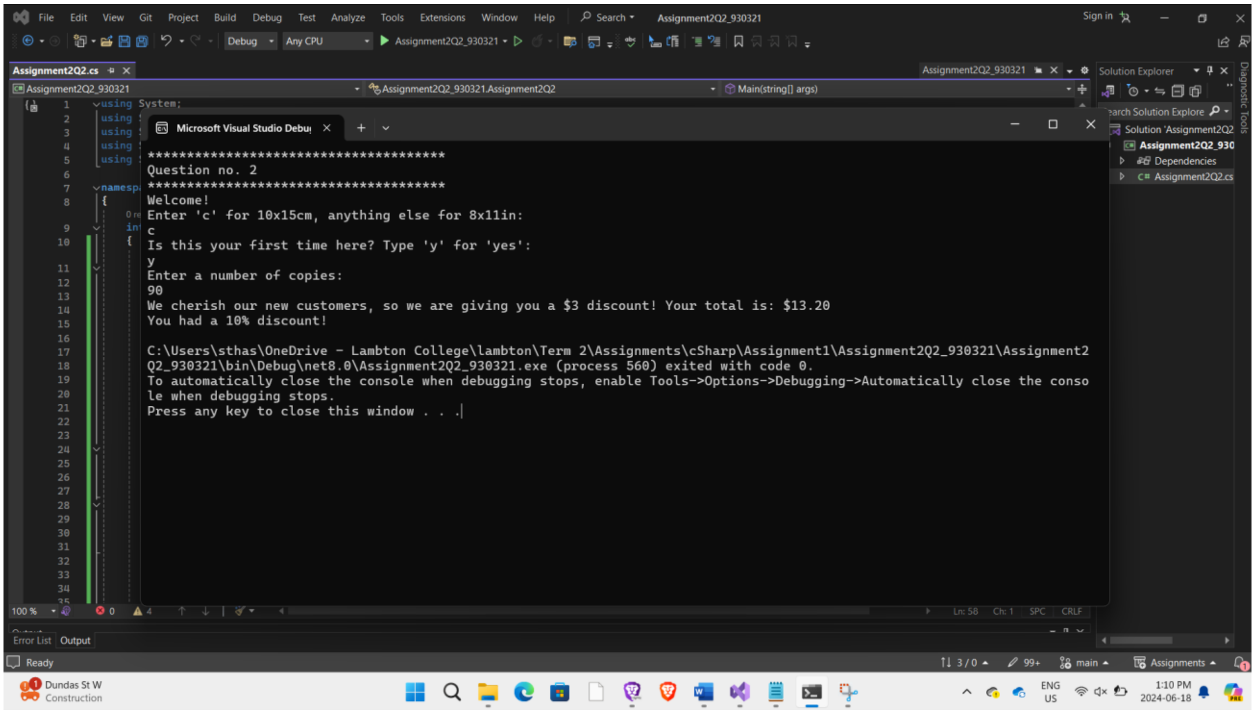


```
            // Asking if it is the first time the customer orders through the company
            Console.WriteLine("Is this your first time here? Type 'y' for 'yes': ");
            string isFirstTimeInput = Console.ReadLine().ToLower();
            bool isFirstTimeCustomer = isFirstTimeInput == "y";

            // Asking for the number of copies
            Console.WriteLine("Enter a number of copies: ");
            int numberOfCopies = int.Parse(Console.ReadLine());

            // Calculate the total cost
            double totalCost = numberOfCopies * costPerPicture;
            bool hasDiscount = false;

            if (numberOfCopies > 50)
            {
                totalCost *= 0.9;
                hasDiscount = true;
            }

            bool hasCoupon = false;
            if (isFirstTimeCustomer && totalCost > 5)
            {
                totalCost -= 3;
                hasCoupon = true;
            }

            Console.WriteLine("We cherish our new customers, so we are giving you a $3 discount! Your total is: $" + totalCost.ToString("F2"));

            if (hasDiscount)
            {
                Console.WriteLine("You had a 10% discount!");
            }

        }
    }
}
```

# Code

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

namespace Assignment2Q2_930321
{
    internal class Assignment2Q2
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("**************************************");
            Console.WriteLine("Question no. 2");
            Console.WriteLine("**************************************");

            Console.WriteLine("Welcome!");

            // asking user to choose a format
            Console.WriteLine("Enter 'c' for 10x15cm, anything else for 8x11in: ");
            string formatChoice = Console.ReadLine().ToLower();
            double costPerPicture = 0;

            if (formatChoice == "c")
            {
                costPerPicture = 0.20;
            }
            else
            {
                costPerPicture = 0.25;
            }

            // Asking if it is the first time the customer orders through the
company
            Console.WriteLine("Is this your first time here? Type 'y' for 'yes': ");
            string isFirstTimeInput = Console.ReadLine().ToLower();
            bool isFirstTimeCustomer = isFirstTimeInput == "y";

            // Asking for the number of copies
            Console.WriteLine("Enter a number of copies: ");
            int numberOfCopies = int.Parse(Console.ReadLine());

            // Calculate the total cost
            double totalCost = numberOfCopies * costPerPicture;
            bool hasDiscount = false;

            if (numberOfCopies > 50)
            {
                totalCost *= 0.9;
                hasDiscount = true;
            }

            bool hasCoupon = false;
            if (isFirstTimeCustomer && totalCost > 5)
            {
                totalCost -= 3;
                hasCoupon = true;
            }

            Console.WriteLine("We cherish our new customers, so we are giving you a
$3 discount! Your total is: $" + totalCost.ToString("F2"));

            if (hasDiscount)
            {
                Console.WriteLine("You had a 10% discount!");
            }

        }
    }
}
```

#Output screenshot



```
**************************************
Question no. 2
**************************************
Welcome!
Enter 'c' for 10x15cm, anything else for 8x11in:
c
Is this your first time here? Type 'y' for 'yes':
y
Enter a number of copies:
90
We cherish our new customers, so we are giving you a $3 discount! Your total is: $13.20
You had a 10% discount!

C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\cSharp\Assignment1\Assignment2Q2_930321\Assignment2
Q2_930321\bin\Debug\net8.0\Assignment2Q2_930321.exe (process 560) exited with code 0.
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 . . .
```