

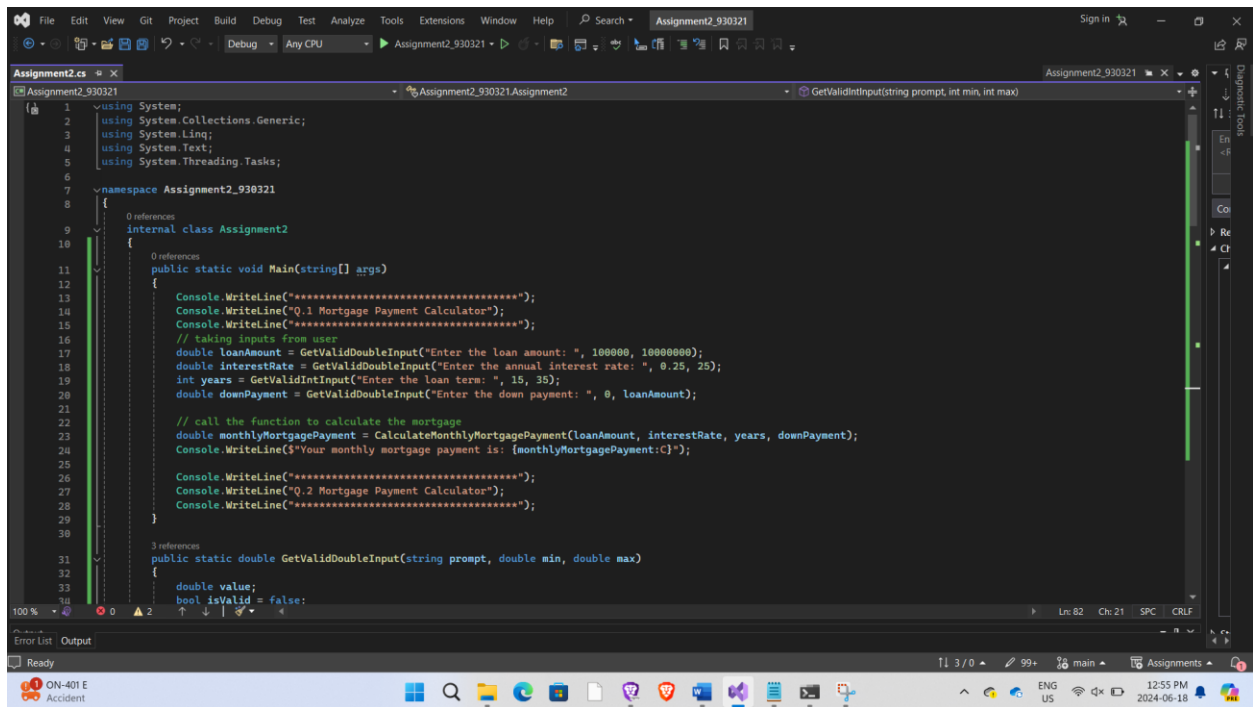
CSharp

Student Name : Shreejana Shrestha

Student Id : C0930321

Question 1

Code screenshot



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Assignment2_930321
8  {
9      0 references
10     internal class Assignment2
11     {
12         0 references
13         public static void Main(string[] args)
14         {
15             Console.WriteLine("*****");
16             Console.WriteLine("Q.1 Mortgage Payment Calculator");
17             Console.WriteLine("*****");
18             // taking inputs from user
19             double loanAmount = GetValidDoubleInput("Enter the loan amount: ", 100000, 100000000);
20             double interestRate = GetValidDoubleInput("Enter the annual interest rate: ", 0.25, 25);
21             int years = GetValidIntInput("Enter the loan term: ", 15, 35);
22             double downPayment = GetValidDoubleInput("Enter the down payment: ", 0, loanAmount);
23
24             // call the function to calculate the mortgage
25             double monthlyMortgagePayment = CalculateMonthlyMortgagePayment(loanAmount, interestRate, years, downPayment);
26             Console.WriteLine($"Your monthly mortgage payment is: {monthlyMortgagePayment:c}");
27
28             Console.WriteLine("*****");
29             Console.WriteLine("Q.2 Mortgage Payment Calculator");
30             Console.WriteLine("*****");
31         }
32     }
33     3 references
34     public static double GetValidDoubleInput(string prompt, double min, double max)
35     {
36         double value;
37         bool isValid = false;
38         while (!isValid)
39         {
40             string input = Console.ReadLine();
41             if (double.TryParse(input, out value) && value >= min && value <= max)
42             {
43                 isValid = true;
44             }
45             else
46             {
47                 Console.WriteLine("Invalid input. Please enter a value between {min} and {max}.");
48             }
49         }
50         return value;
51     }
52 }
```

This screenshot shows the Visual Studio IDE with the file `Assignment2_930321.cs` open. The code defines a method `GetValidDoubleInput` that takes a string prompt and two double values (min and max) as parameters. It uses a `do-while` loop to repeatedly prompt the user for a double value until it is valid (i.e., within the specified range). The code is as follows:

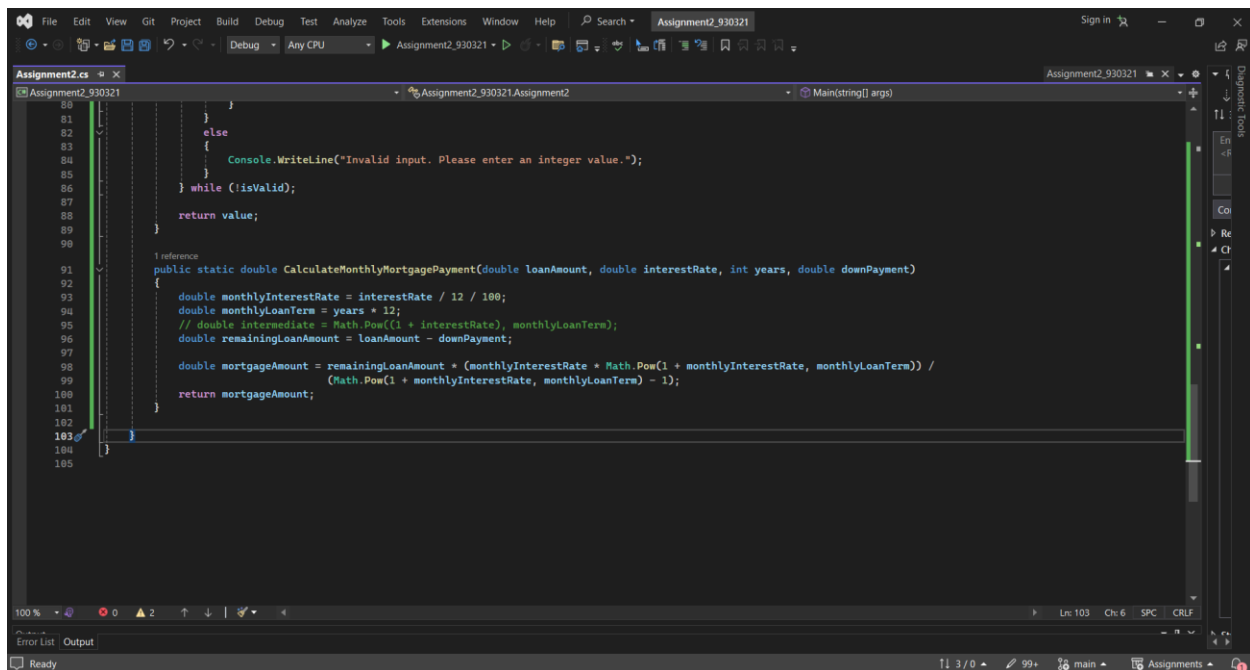
```
31 3 references
32 public static double GetValidDoubleInput(string prompt, double min, double max)
33 {
34     double value;
35     bool isValid = false;
36
37     do
38     {
39         Console.Write(prompt);
40         string input = Console.ReadLine();
41
42         if (double.TryParse(input, out value))
43         {
44             if (value >= min && value <= max)
45             {
46                 isValid = true;
47             }
48             else
49             {
50                 Console.WriteLine($"Value must be between {min} and {max}.");
51             }
52         }
53         else
54         {
55             Console.WriteLine("Invalid input. Please enter a numeric value.");
56         }
57     } while (!isValid);
58
59     return value;
60
61 1 reference
62 public static int GetValidIntInput(string prompt, int min, int max)
63 {
64     int value;
65     bool isValid = false;
```

The status bar at the bottom indicates the file is ready, and the taskbar shows the system clock as 12:56 PM on 2024-06-18.

This screenshot shows the Visual Studio IDE with the file `Assignment2_930321.cs` open. The code defines two methods: `GetValidIntInput` and `CalculateMonthlyMortgagePayment`. The `GetValidIntInput` method is similar to the one in the first screenshot, but it prompts for an integer value. The `CalculateMonthlyMortgagePayment` method takes a double loan amount, a double interest rate, an integer number of years, and a double down payment as parameters. It calculates the monthly interest rate and the monthly loan term. The code is as follows:

```
61 1 reference
62 public static int GetValidIntInput(string prompt, int min, int max)
63 {
64     int value;
65     bool isValid = false;
66
67     do
68     {
69         Console.Write(prompt);
70         string input = Console.ReadLine();
71
72         if (int.TryParse(input, out value))
73         {
74             if (value >= min && value <= max)
75             {
76                 isValid = true;
77             }
78             else
79             {
80                 Console.WriteLine($"Value must be between {min} and {max}.");
81             }
82         }
83         else
84         {
85             Console.WriteLine("Invalid input. Please enter an integer value.");
86         }
87     } while (!isValid);
88
89     return value;
90
91 1 reference
92 public static double CalculateMonthlyMortgagePayment(double loanAmount, double interestRate, int years, double downPayment)
93 {
94     double monthlyInterestRate = interestRate / 12 / 100;
95     double monthlyLoanTerm = years * 12;
```

The status bar at the bottom indicates the file is ready, and the taskbar shows the system clock as 12:56 PM on 2024-06-18.



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);

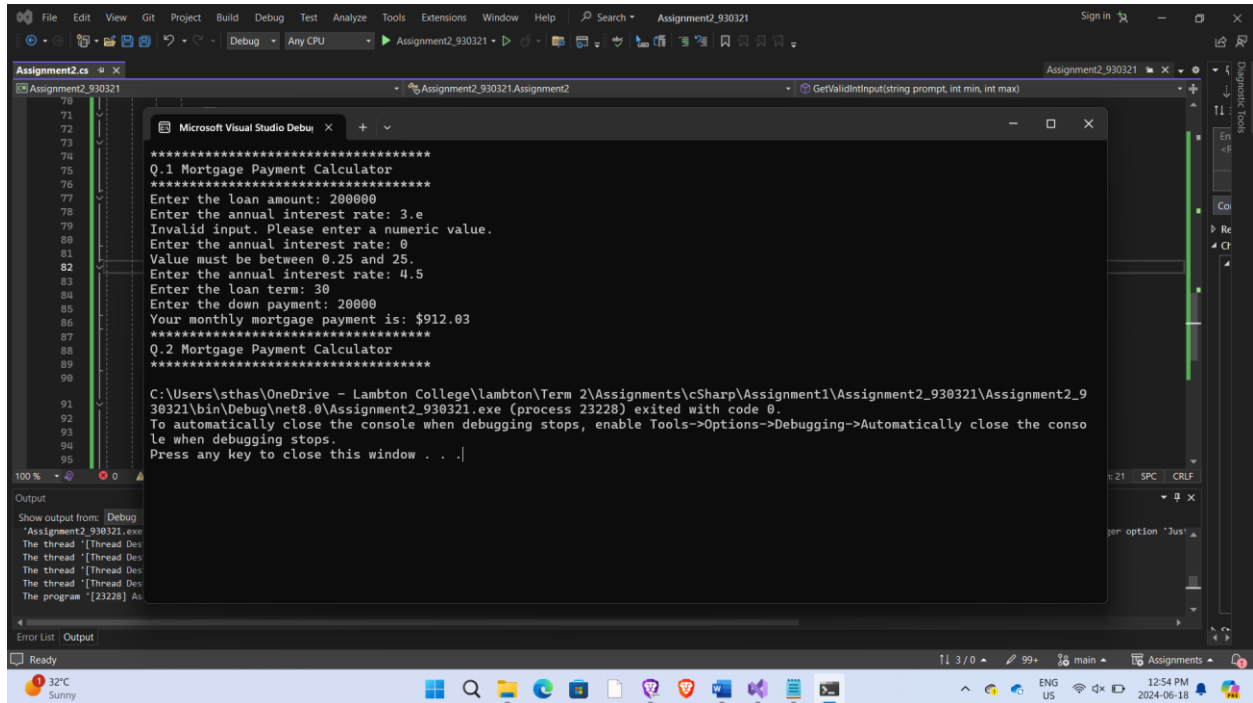
return value;
}

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



```
*****
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_930321\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 console when debugging stops.
Press any key to close this window . . .|
```

Output

Show output from: Debug

Assignment2_930321.exe

The thread '[Thread Des

The thread '[Thread Des

The thread '[Thread Des

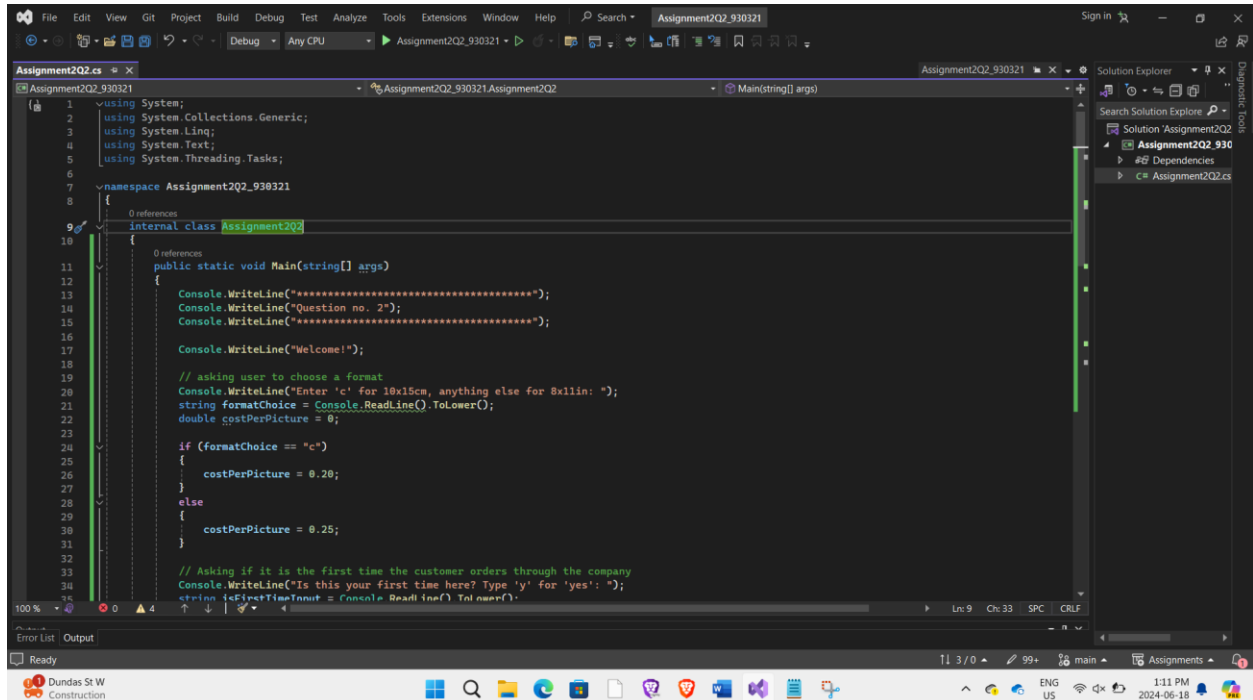
The thread '[Thread Des

The thread '[Thread Des

The program '[23228] As

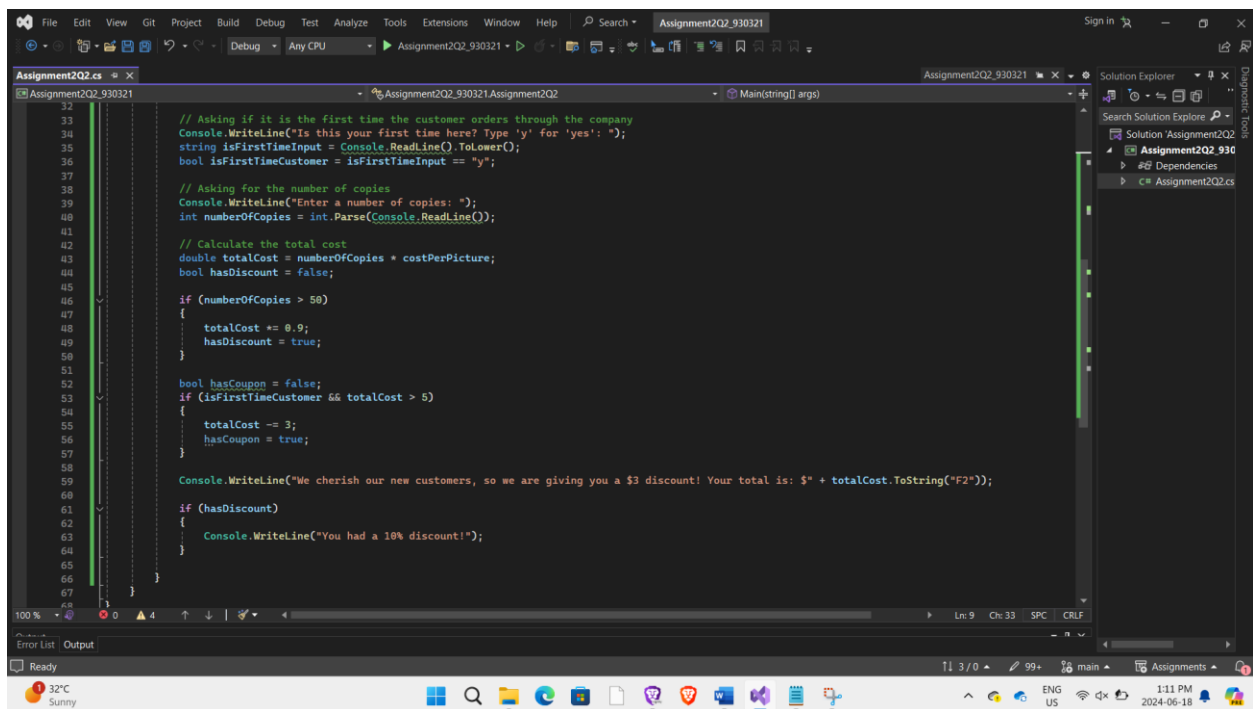
Question 2

Code screenshot



This screenshot shows the first part of the C# code in the file `Assignment2Q2_930321.cs`. The code defines an internal class `Assignment2Q2` with a `Main` method. The `Main` method starts by writing a header, then a welcome message, and prompts the user to choose a format ('c' for 10x15cm, anything else for 8x11in). It then reads the user's choice and sets the `costPerPicture` accordingly (0.20 for 'c', 0.25 otherwise).

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Assignment2Q2_930321
8 {
9     internal class Assignment2Q2
10     {
11         public static void Main(string[] args)
12         {
13             Console.WriteLine("*****");
14             Console.WriteLine("Question no. 2*");
15             Console.WriteLine("*****");
16
17             Console.WriteLine("Welcome!");
18
19             // asking user to choose a format
20             Console.WriteLine("Enter 'c' for 10x15cm, anything else for 8x11in: ");
21             string formatChoice = Console.ReadLine().ToLower();
22             double costPerPicture = 0;
23
24             if (formatChoice == "c")
25             {
26                 costPerPicture = 0.20;
27             }
28             else
29             {
30                 costPerPicture = 0.25;
31             }
32
33             // Asking if it is the first time the customer orders through the company
34             Console.WriteLine("Is this your first time here? Type 'y' for 'yes': ");
35             string isFirstTimeInput = Console.ReadLine().ToLower();
36             bool isFirstTimeCustomer = isFirstTimeInput == "y";
```



This screenshot shows the second part of the C# code, continuing from the previous section. It calculates the total cost based on the number of copies and the cost per picture. It also checks for a discount: a 10% discount for first-time customers with a total cost greater than \$5, and a \$3 discount for all customers with a total cost greater than \$50. The final total cost is printed, and a message about the discount is shown if applicable.

```
37
38 // Asking for the number of copies
39 Console.WriteLine("Enter a number of copies: ");
40 int numberOfCopies = int.Parse(Console.ReadLine());
41
42 // Calculate the total cost
43 double totalCost = numberOfCopies * costPerPicture;
44 bool hasDiscount = false;
45
46 if (numberOfCopies > 50)
47 {
48     totalCost -= 0.9;
49     hasDiscount = true;
50 }
51
52 bool hasCoupon = false;
53 if (isFirstTimeCustomer && totalCost > 5)
54 {
55     totalCost -= 3;
56     hasCoupon = true;
57 }
58
59 Console.WriteLine("We cherish our new customers, so we are giving you a $3 discount! Your total is: $" + totalCost.ToString("F2"));
60
61 if (hasDiscount)
62 {
63     Console.WriteLine("You had a 10% discount!");
64 }
65
66 }
67 }
```

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

```

1  using System;
2  using Microsoft.VisualBasic;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7  namespace Assignment2Q2_930321
8  {
9      Welcome!
10     Enter 'c' for 10x15cm, anything else for 8x11in:
11     c
12     Is this your first time here? Type 'y' for 'yes':
13     y
14     Enter a number of copies:
15     90
16     We cherish our new customers, so we are giving you a $3 discount! Your total is: $13.20
17     You had a 10% discount!
18
19     C:\Users\sthas\OneDrive - Lambton College\Lambton\Term 2\Assignments\cSharp\Assignment1\Assignment2Q2_930321\Assignment2
20     Q2_930321\bin\Debug\net8.0\Assignment2Q2_930321.exe (process 568) exited with code 0.
21     To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console
22     when debugging stops.
23     Press any key to close this window . . .
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
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

```