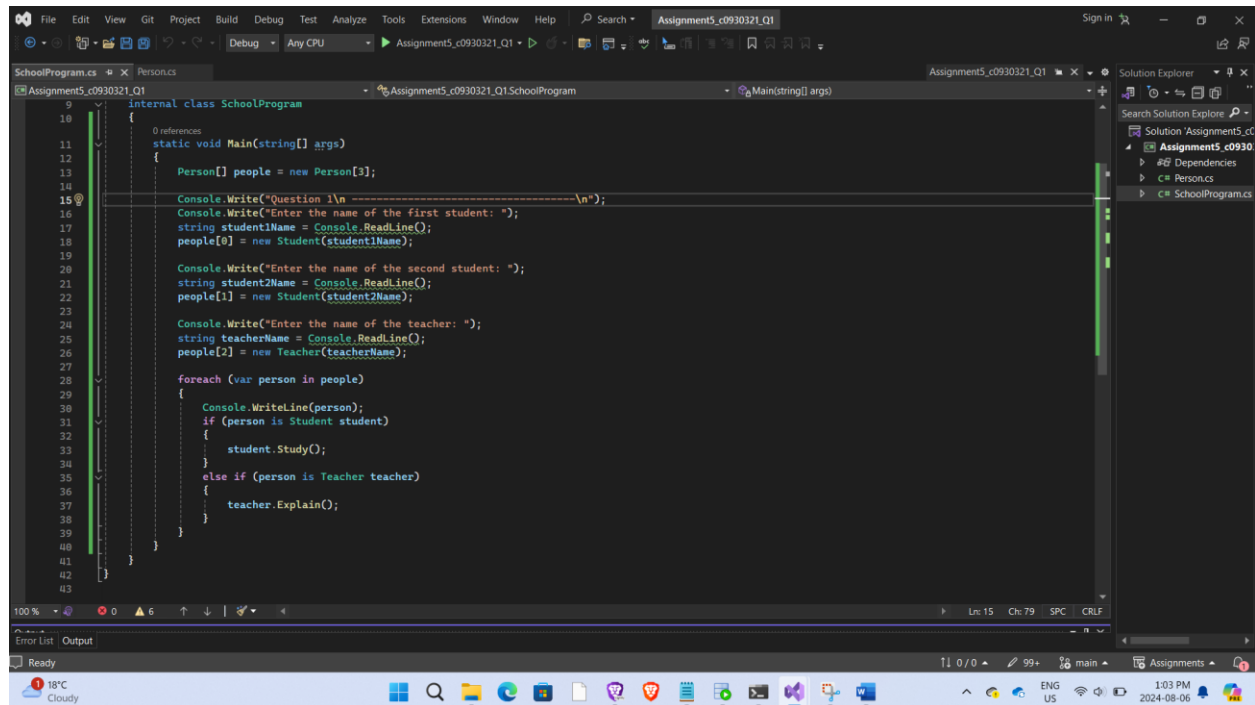


Question 1:

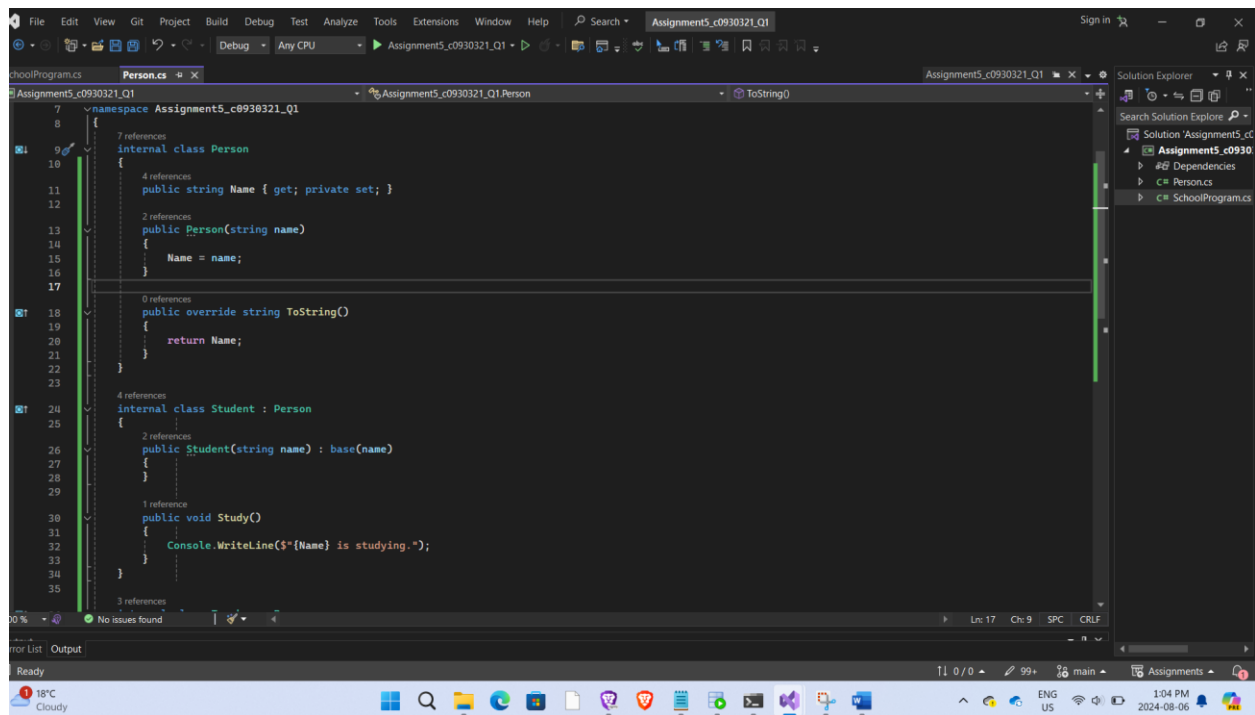
Code screenshot



The screenshot shows the Visual Studio IDE with a C# project named 'Assignment5_c0930321_Q1'. The code is in a file named 'SchoolProgram.cs' and is part of an 'internal class SchoolProgram'. The code defines a 'Main' method that takes an array of arguments. It prompts the user to enter the names of two students and one teacher, then creates 'Student' and 'Teacher' objects. A 'foreach' loop iterates over the 'people' array, calling 'Study()' on students and 'Explain()' on teachers. The code is as follows:

```
9 internal class SchoolProgram
10 {
11     0 references
12     static void Main(string[] args)
13     {
14         Person[] people = new Person[3];
15         Console.WriteLine("Question 1\n ----- \n");
16         Console.WriteLine("Enter the name of the first student: ");
17         string student1Name = Console.ReadLine();
18         people[0] = new Student(student1Name);
19
20         Console.WriteLine("Enter the name of the second student: ");
21         string student2Name = Console.ReadLine();
22         people[1] = new Student(student2Name);
23
24         Console.WriteLine("Enter the name of the teacher: ");
25         string teacherName = Console.ReadLine();
26         people[2] = new Teacher(teacherName);
27
28         foreach (var person in people)
29         {
30             Console.WriteLine(person);
31             if (person is Student student)
32             {
33                 student.Study();
34             }
35             else if (person is Teacher teacher)
36             {
37                 teacher.Explain();
38             }
39         }
40     }
41 }
42
43
```

The bottom of the image shows the Windows taskbar with the system clock at 1:03 PM on 2024-08-06.



Code

using System;

```
namespace Assignment5_c0930321_Q1
```

```
{
```

```
    internal class Person
```

```
    {
```

```
        public string Name { get; private set; }
```

```
        public Person(string name)
```

```
        {
```

```
            Name = name;
```

```
        }
```

```
public override string ToString()
{
    return Name;
}
}
```

```
internal class Student : Person
{
    public Student(string name) : base(name)
    {
    }

    public void Study()
    {
        Console.WriteLine($"{Name} is studying.");
    }
}
```

```
internal class Teacher : Person
{
    public Teacher(string name) : base(name)
    {
    }

    public void Explain()
    {
        Console.WriteLine($"{Name} is explaining.");
    }
}
```

```
    }  
  }  
}
```

```
using System;
```

```
namespace Assignment5_c0930321_Q1
```

```
{  
    internal class SchoolProgram  
    {  
        static void Main(string[] args)  
        {  
            Person[] people = new Person[3];  
  
            Console.WriteLine("Question 1\n -----\\n");  
            Console.WriteLine("Enter the name of the first student: ");  
            string student1Name = Console.ReadLine();  
            people[0] = new Student(student1Name);  
  
            Console.WriteLine("Enter the name of the second student: ");  
            string student2Name = Console.ReadLine();  
            people[1] = new Student(student2Name);  
  
            Console.WriteLine("Enter the name of the teacher: ");  
            string teacherName = Console.ReadLine();
```

```
people[2] = new Teacher(teacherName);
```

```
foreach (var person in people)
```

```
{
```

```
    Console.WriteLine(person);
```

```
    if (person is Student student)
```

```
    {
```

```
        student.Study();
```

```
    }
```

```
    else if (person is Teacher teacher)
```

```
    {
```

```
        teacher.Explain();
```

```
    }
```

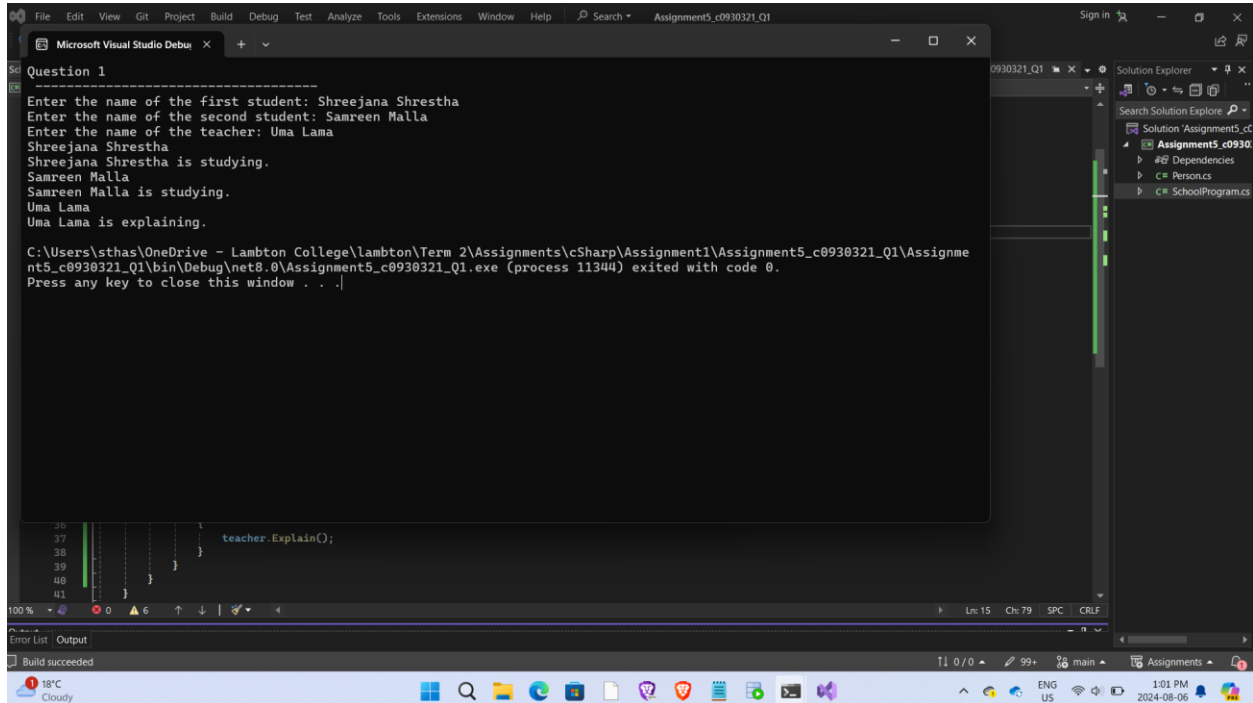
```
}
```

```
}
```

```
}
```

```
}
```

Output Screenshot



```
Question 1
-----
Enter the name of the first student: Shreejana Shrestha
Enter the name of the second student: Samreen Malla
Enter the name of the teacher: Uma Lama
Shreejana Shrestha
Shreejana Shrestha is studying.
Samreen Malla
Samreen Malla is studying.
Uma Lama
Uma Lama is explaining.

C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\cSharp\Assignment1\Assignment5_c0930321_Q1\Assignme
nt5_c0930321_Q1\bin\Debug\net8.0\Assignment5_c0930321_Q1.exe (process 11344) exited with code 0.
Press any key to close this window . . .|

36 |
37 |         teacher.Explain();
38 |     }
39 | }
40 |
41 |
```

Question 2:

Code:

using System;

namespace Assignment5_Q2_c0930321

{

public abstract class Shape

{

protected Location c;

public Shape(Location c)

```

{
    this.c = c;
}

public abstract double Area();
public abstract double Perimeter();

public override string ToString()
{
    return $"Shape at location ({c.x}, {c.y})\n";
}
}

public class Rectangle : Shape
{
    private double side1;
    private double side2;

    public Rectangle(Location c, double side1, double side2) : base(c)
    {
        this.side1 = side1;
        this.side2 = side2;
    }

    public override double Area()
    {
        return side1 * side2;
    }
}

```

```
}
```

```
public override double Perimeter()
```

```
{
```

```
    return 2 * (side1 + side2);
```

```
}
```

```
public override string ToString()
```

```
{
```

```
    return $"Rectangle at location ({c.x}, {c.y}) with sides {side1} and {side2}\n";
```

```
}
```

```
}
```

```
public class Circle : Shape
```

```
{
```

```
    private double radius;
```

```
    public Circle(Location c, double radius) : base(c)
```

```
{
```

```
        this.radius = radius;
```

```
}
```

```
public override double Area()
```

```
{
```

```
    return Math.PI * radius * radius;
```

```
}
```



```
        public override double Perimeter()
        {
            return 2 * Math.PI * radius;
        }

        public override string ToString()
        {
            return $"Circle at location ({c.x}, {c.y}) with radius {radius}\n";
        }
    }
}
```

```
using System;
```

```
namespace Assignment5_Q2_c0930321
```

```
{
    public class Location
    {
        public double x { get; set; }
        public double y { get; set; }

        public Location(double x, double y)
        {
            this.x = x;
            this.y = y;
        }
    }
}
```

```

    }
}

using System;

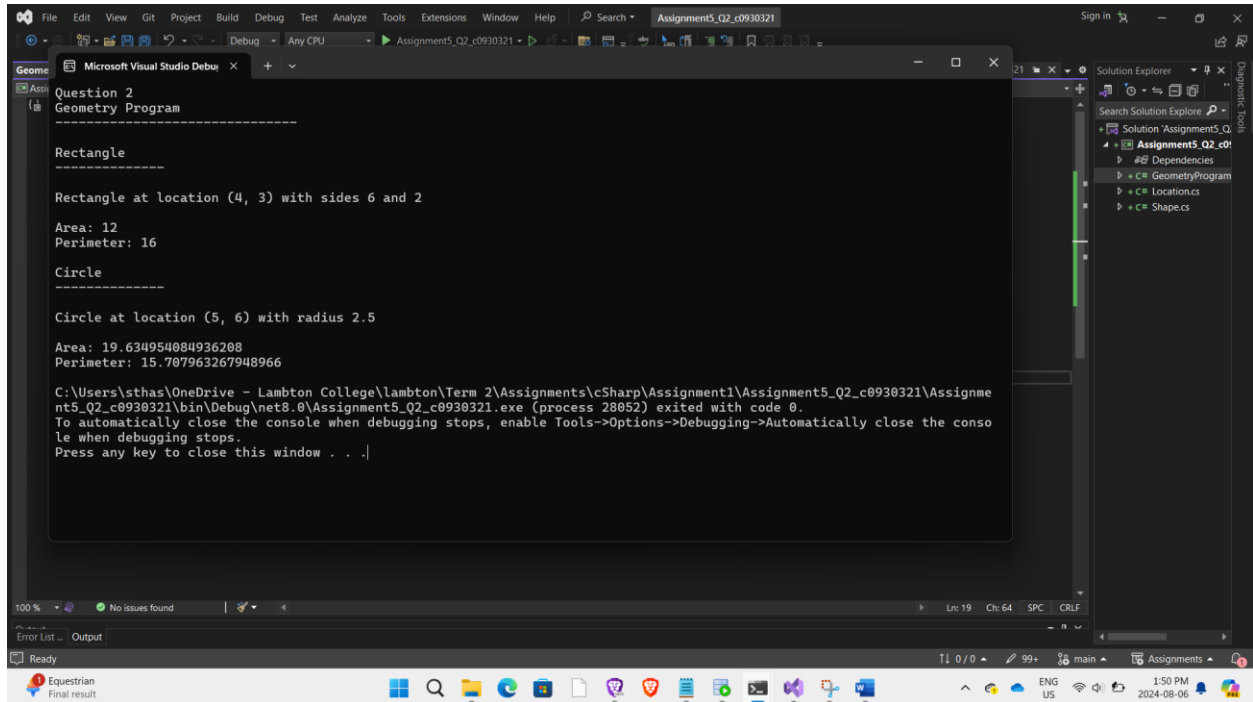
namespace Assignment5_Q2_c0930321
{
    internal class GeometryProgram
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Question 2\nGeometry Program\n-----\n");

            Location location1 = new Location(4.0, 3.0);
            Console.WriteLine("Rectangle\n-----\n");
            Shape rect = new Rectangle(location1, 6.0, 2.0);
            Console.WriteLine(rect.ToString());
            Console.WriteLine($"Area: {rect.Area()}");
            Console.WriteLine($"Perimeter: {rect.Perimeter()}\n");

            Location location2 = new Location(5.0, 6.0);
            Console.WriteLine("Circle\n-----\n");
            Shape circle = new Circle(location2, 2.5);
            Console.WriteLine(circle.ToString());
            Console.WriteLine($"Area: {circle.Area()}");
            Console.WriteLine($"Perimeter: {circle.Perimeter()}");
        }
    }
}

```

Output Screenshot



```
Question 2
Geometry Program
-----

Rectangle
-----

Rectangle at location (4, 3) with sides 6 and 2

Area: 12
Perimeter: 16

Circle
-----

Circle at location (5, 6) with radius 2.5

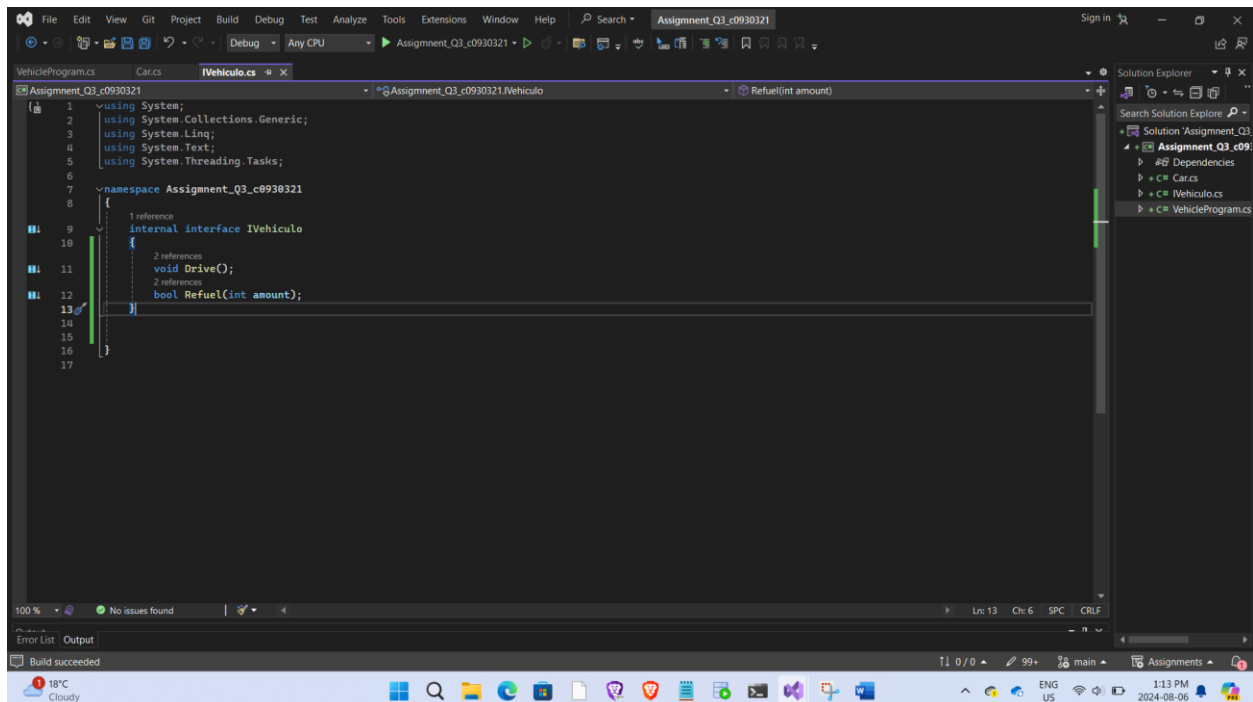
Area: 19.634954084936208
Perimeter: 15.707963267948966

C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\cSharp\Assignment1\Assignment5_Q2_c0930321\Assignme
nt5_Q2_c0930321\bin\Debug\net8.0\Assignment5_Q2_c0930321.exe (process 28052) 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 . . .|
```

The screenshot shows the Microsoft Visual Studio IDE. The main window displays the output of a C# program. The output shows calculations for a rectangle and a circle. The rectangle is located at (4, 3) with sides 6 and 2, resulting in an area of 12 and a perimeter of 16. The circle is located at (5, 6) with a radius of 2.5, resulting in an area of 19.634954084936208 and a perimeter of 15.707963267948966. The console window also shows the path to the executable file and a message indicating that the process exited with code 0. The Solution Explorer on the right shows the project structure, including the 'Assignment5_Q2_c0930321' project and its subfolders 'Geometry' and 'Shape'.

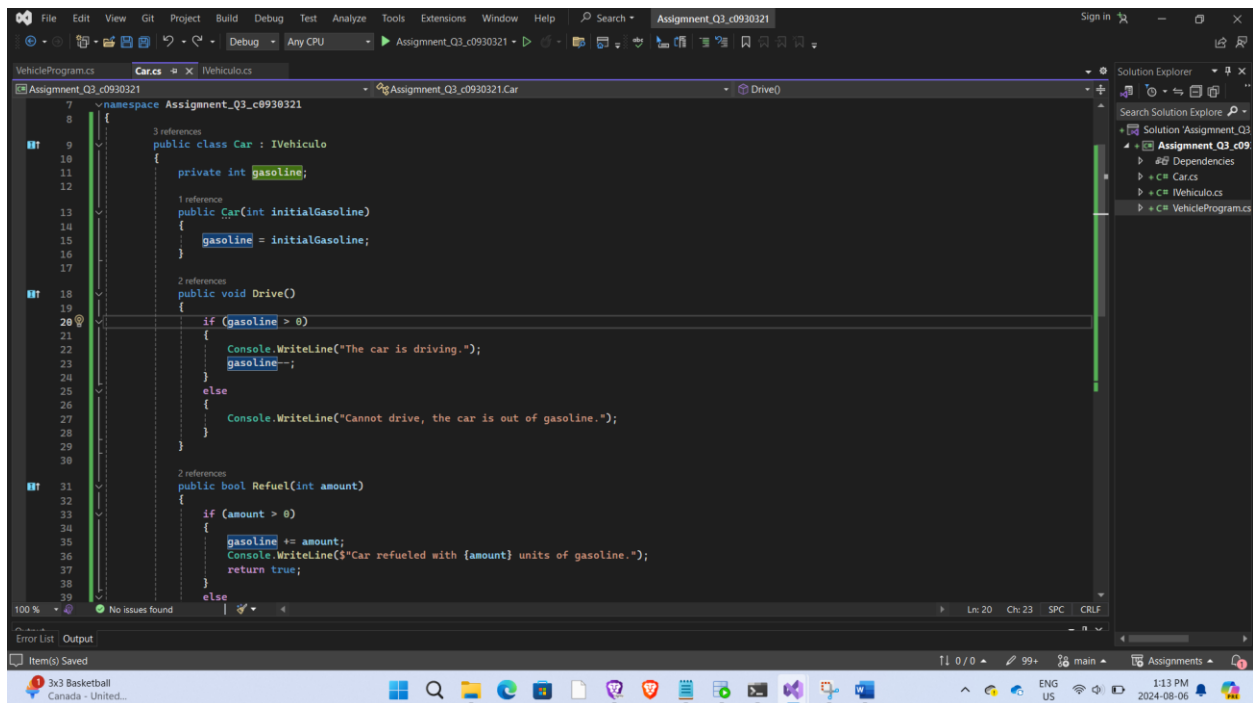
Question 3:

Code screenshot



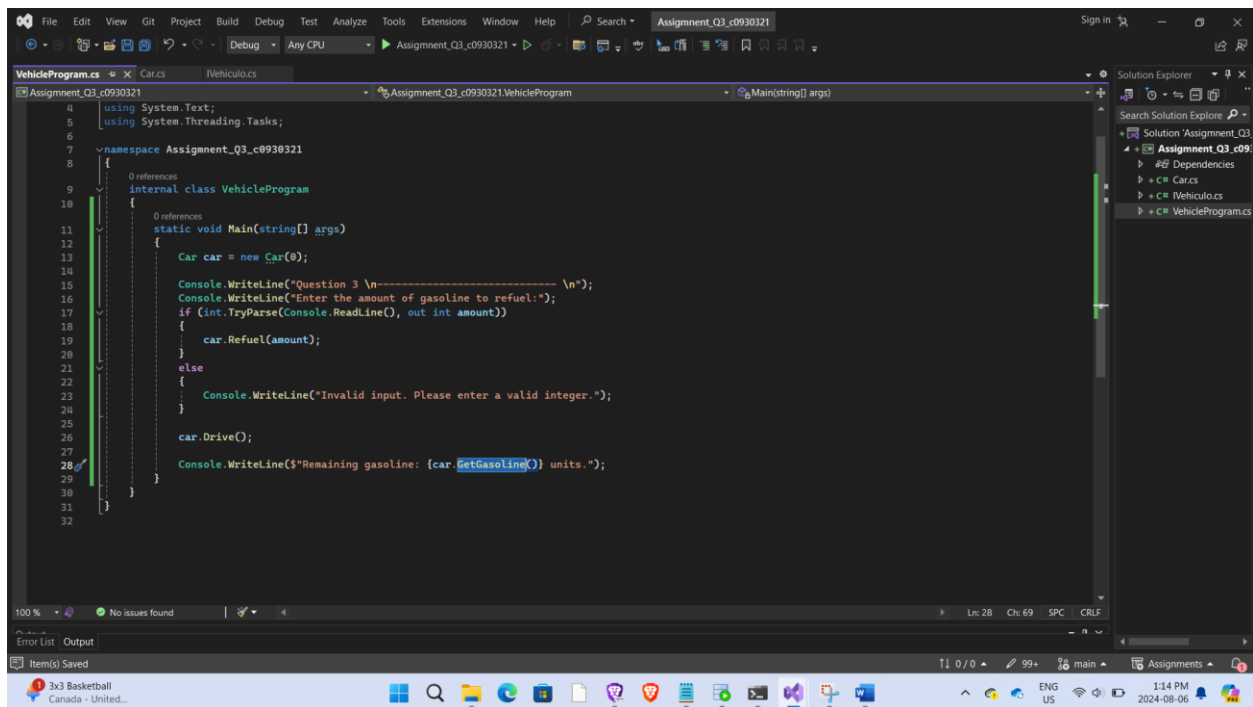
This screenshot shows the Visual Studio IDE with the file `IVehicle.cs` open. The code defines an internal interface `IVehicle` within the namespace `Assignment_Q3_c0930321`. The interface includes two methods: `void Drive()` and `bool Refuel(int amount)`. The Solution Explorer on the right shows the project structure, including `Assignment_Q3_c0930321`, `Car.cs`, `IVehicle.cs`, and `VehicleProgram.cs`. The status bar at the bottom indicates "Build succeeded".

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Assignment_Q3_c0930321
8 {
9     1 reference
10     internal interface IVehicle
11     {
12         2 references
13         void Drive();
14         2 references
15         bool Refuel(int amount);
16     }
17 }
```



This screenshot shows the Visual Studio IDE with the file `Car.cs` open. The code implements the `IVehicle` interface. It defines a `Car` class with a private `gasoline` property. The `Car` class has a constructor `Car(int initialGasoline)` that initializes `gasoline`. It also implements the `Drive()` and `Refuel(int amount)` methods. The `Drive()` method checks if `gasoline` is greater than 0; if so, it prints "The car is driving." and decrements `gasoline`. The `Refuel()` method checks if `amount` is greater than 0; if so, it increments `gasoline` by `amount` and prints a message. The Solution Explorer on the right shows the project structure, including `Assignment_Q3_c0930321`, `Car.cs`, `IVehicle.cs`, and `VehicleProgram.cs`. The status bar at the bottom indicates "No issues found".

```
7 namespace Assignment_Q3_c0930321
8 {
9     3 references
10     public class Car : IVehicle
11     {
12         private int gasoline;
13
14         1 reference
15         public Car(int initialGasoline)
16         {
17             gasoline = initialGasoline;
18         }
19
20         2 references
21         public void Drive()
22         {
23             if (gasoline > 0)
24             {
25                 Console.WriteLine("The car is driving.");
26                 gasoline--;
27             }
28             else
29             {
30                 Console.WriteLine("Cannot drive, the car is out of gasoline.");
31             }
32         }
33
34         2 references
35         public bool Refuel(int amount)
36         {
37             if (amount > 0)
38             {
39                 gasoline += amount;
40                 Console.WriteLine($"Car refueled with {amount} units of gasoline.");
41                 return true;
42             }
43             else
44             {
45                 return false;
46             }
47         }
48     }
49 }
```



Code

using System;

namespace Assignmnet_Q3_c0930321

{

internal class VehicleProgram

{

static void Main(string[] args)

{

Car car = new Car(0);

Console.WriteLine("Question 3 \n----- \n");

```

        Console.WriteLine("Enter the amount of gasoline to refuel:");
        if (int.TryParse(Console.ReadLine(), out int amount))
        {
            car.Refuel(amount);
        }
        else
        {
            Console.WriteLine("Invalid input. Please enter a valid integer.");
        }

        car.Drive();

        Console.WriteLine($"Remaining gasoline: {car.GetGasoline()} units.");
    }
}

```

```

using System;

```

```

namespace Assignment_Q3_c0930321
{
    public class Car : IVehiculo
    {
        private int gasoline;
    }
}

```

```
public Car(int initialGasoline)
{
    gasoline = initialGasoline;
}
```

```
public void Drive()
{
    if (gasoline > 0)
    {
        Console.WriteLine("The car is driving.");
        gasoline--;
    }
    else
    {
        Console.WriteLine("Cannot drive, the car is out of gasoline.");
    }
}
```

```
public bool Refuel(int amount)
{
    if (amount > 0)
    {
        gasoline += amount;
        Console.WriteLine($"Car refueled with {amount} units of gasoline.");
        return true;
    }
    else
```

```

        {
            Console.WriteLine("Invalid amount of gasoline to refuel.");
            return false;
        }
    }

    public int GetGasoline()
    {
        return gasoline;
    }
}

```

```
using System;
```

```
namespace Assignment_Q3_c0930321
```

```

{
    internal interface IVehiculo
    {
        void Drive();
        bool Refuel(int amount);
    }
}

```


Output Screenshot

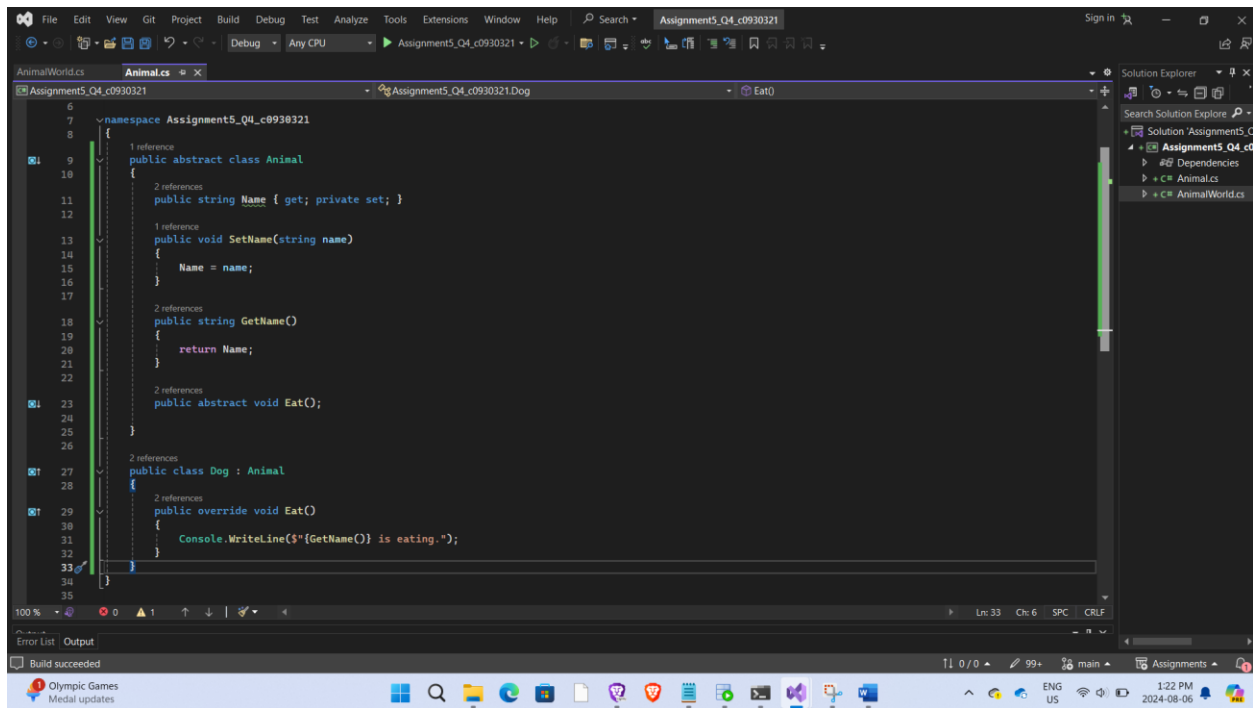
The screenshot shows the Microsoft Visual Studio IDE with a C# project named 'Assignment_Q3_c0930321'. The 'Output' window displays the following text:

```
Question 3
-----
Enter the amount of gasoline to refuel:
633
Car refueled with 633 units of gasoline.
The car is driving.
Remaining gasoline: 632 units.
C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\cSharp\Assignment1\Assignment_Q3_c0930321\Assignmen
t_Q3_c0930321\bin\Debug\net8.0\Assignment_Q3_c0930321.exe (process 25888) exited with code 0.
Press any key to close this window . . .
```

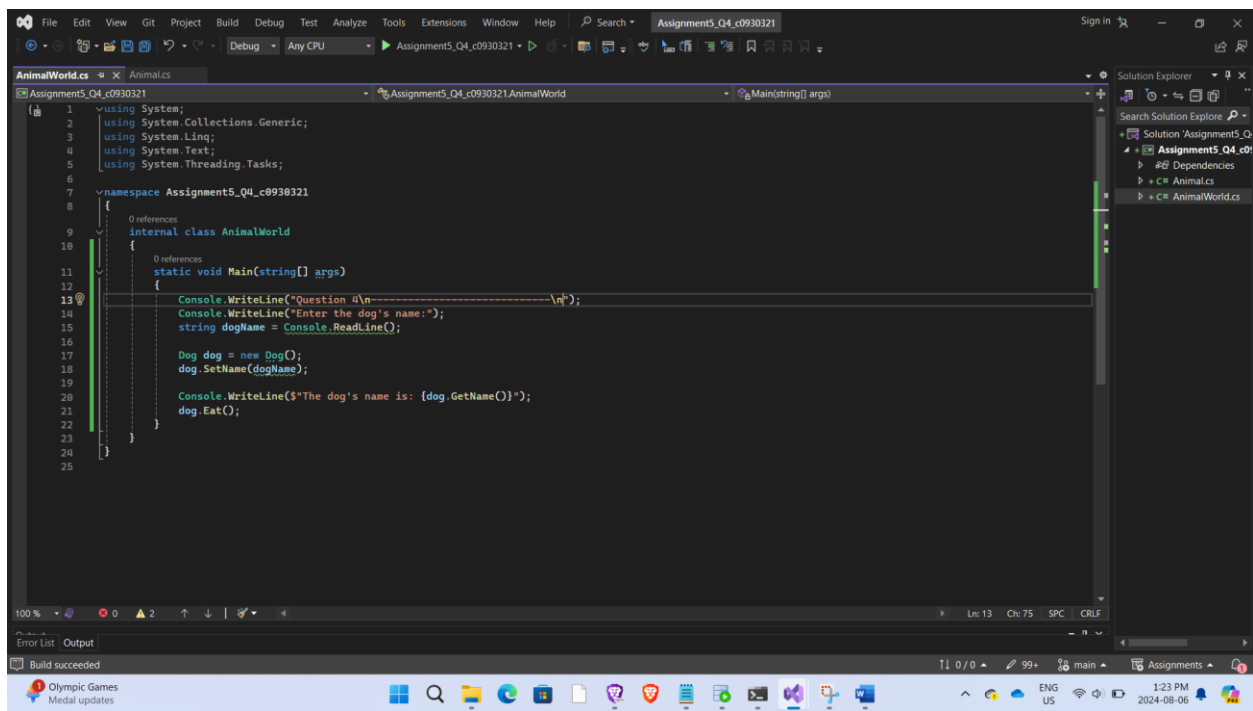
The 'Solution Explorer' on the right shows the project structure with files: 'Assignment_Q3_c0930321.cs', 'Car.cs', 'IVehicle.cs', and 'VehicleProgram.cs'. The 'Error List' at the bottom shows 'Build succeeded'.

Question 4:

Code screenshot



```
6 namespace Assignment5_Q4_c0930321
7 {
8     1 reference
9     public abstract class Animal
10     {
11         2 references
12         public string Name { get; private set; }
13
14         1 reference
15         public void SetName(string name)
16         {
17             Name = name;
18         }
19
20         2 references
21         public string GetName()
22         {
23             return Name;
24         }
25
26         2 references
27         public abstract void Eat();
28     }
29
30     2 references
31     public class Dog : Animal
32     {
33         2 references
34         public override void Eat()
35         {
36             Console.WriteLine($"{GetName()} is eating.");
37         }
38     }
39 }
```



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Assignment5_Q4_c0930321
8 {
9     0 references
10     internal class AnimalWorld
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             Console.WriteLine("Question 4\n");
16             Console.WriteLine("Enter the dog's name:");
17             string dogName = Console.ReadLine();
18
19             Dog dog = new Dog();
20             dog.SetName(dogName);
21
22             Console.WriteLine($"The dog's name is: {dog.GetName()}");
23             dog.Eat();
24         }
25     }
26 }
```

Code

```
using System;
```

```
namespace Assignment5_Q4_c0930321
```

```
{
```

```
    internal class AnimalWorld
```

```
    {
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Console.WriteLine("Question 4\n-----\n");
```

```
            Console.WriteLine("Enter the dog's name:");
```

```
            string dogName = Console.ReadLine();
```

```
            Dog dog = new Dog();
```

```
            dog.SetName(dogName);
```

```
            Console.WriteLine($"The dog's name is: {dog.GetName()}");
```

```
            dog.Eat();
```

```
        }
```

```
    }
```

```
}
```

```
using System;
```

```
namespace Assignment5_Q4_c0930321
```

```
{
```

```
public abstract class Animal
{
    public string Name { get; private set; }

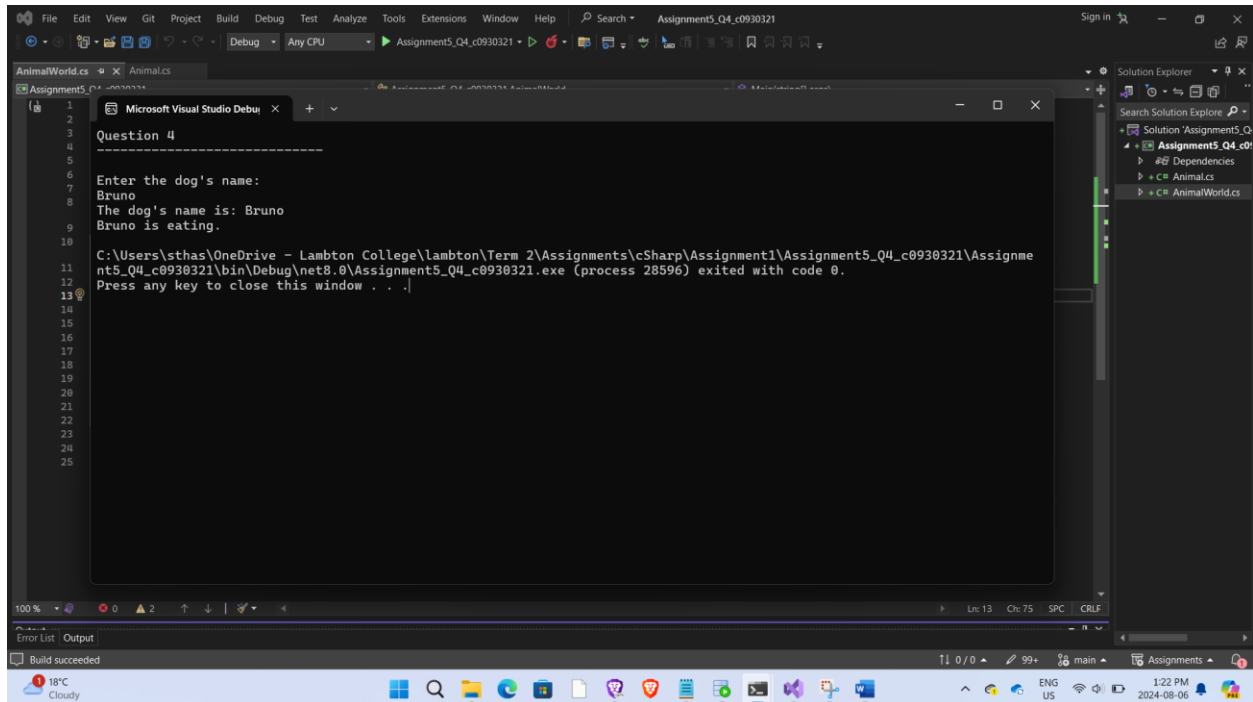
    public void SetName(string name)
    {
        Name = name;
    }

    public string GetName()
    {
        return Name;
    }

    public abstract void Eat();
}

public class Dog : Animal
{
    public override void Eat()
    {
        Console.WriteLine($"{GetName()} is eating.");
    }
}
}
```

Output Screenshot



Question 5:

Code:

using System;

namespace Assignment5_Q5_c0930321

{

internal class Student

{

private static double serviceTax = 12.3;

public static double ServiceTax

{

get { return serviceTax; }

```
    set { serviceTax = value; }  
}
```

```
public int RollNo { get; private set; }  
public string Name { get; private set; }  
public Course CourseJoined { get; private set; }  
public double FeePaid { get; private set; }
```

```
public Student(int rollNo, string name, Course course)  
{  
    RollNo = rollNo;  
    Name = name;  
    CourseJoined = course;  
    FeePaid = 0;  
}
```

```
public void Payment(double amount)  
{  
    FeePaid += amount;  
}
```

```
public double DueAmount  
{  
    get { return TotalFee - FeePaid; }  
}
```

```
public double TotalFee
```

```

    {
        get { return CourseJoined.GetTotalFee() * (1 + ServiceTax / 100); }
    }

    public void Print()
    {
        Console.WriteLine($"Roll No: {RollNo}\n, Name: {Name}\n, Course:
{CourseJoined.Name}\n, Fee Paid: {FeePaid:C}\n, Due Amount: {DueAmount:C}\n");
    }
}

using System;

namespace Assignment5_Q5_c0930321
{
    public abstract class Course
    {
        public string Name { get; private set; }
        public int Duration { get; private set; }
        public double CourseFee { get; private set; }

        public Course(string name, int duration, double courseFee)
        {
            Name = name;
            Duration = duration;
            CourseFee = courseFee;
        }
    }
}

```

```
}
```

```
public abstract double GetTotalFee();
```

```
public virtual void Print()
```

```
{
```

```
    Console.WriteLine($"Course: {Name}, Duration: {Duration} months, Course Fee:  
{CourseFee:C}\n");
```

```
}
```

```
}
```

```
class PartTimeCourse : Course
```

```
{
```

```
    public string Timing { get; private set; }
```

```
    public PartTimeCourse(string name, int duration, double courseFee, string timing)
```

```
        : base(name, duration, courseFee)
```

```
{
```

```
    Timing = timing;
```

```
}
```

```
public override double GetTotalFee()
```

```
{
```

```
    return CourseFee * 0.9;
```

```
}
```

```
public override void Print()
```



```
{  
    base.Print();  
    Console.WriteLine($"Timing: {Timing}");  
}  
}
```

```
class OnsiteCourse : Course  
{  
    public string CompanyName { get; private set; }  
    public int NumberOfCandidates { get; private set; }  
  
    public OnsiteCourse(string name, int duration, double courseFee, string  
companyName, int numberOfCandidates)  
        : base(name, duration, courseFee)  
    {  
        CompanyName = companyName;  
        NumberOfCandidates = numberOfCandidates;  
    }  
  
    public override double GetTotalFee()  
    {  
        return CourseFee * 1.1;  
    }  
  
    public override void Print()  
    {  
        base.Print();  
    }  
}
```

```

        Console.WriteLine($"Company: {CompanyName},\n Number of Candidates:
{NumberOfCandidates}\n");
    }

```

```

    }
}

```

```

using System;

```

```

namespace Assignment5_Q5_c0930321

```

```

{

```

```

    internal class StudentCourseMgmt

```

```

    {

```

```

        static void Main(string[] args)

```

```

        {

```

```

            Console.WriteLine("Question 5\n Student Course Management\n-----
-----\n");

```

```

            // Creating courses

```

```

            Course csharpCourse = new PartTimeCourse("C#", 6, 2000, "Evening");

```

```

            Course aspNetCourse = new OnsiteCourse("ASP.NET", 4, 3000, "ABC Corp",
10);

```

```

            Course pythonCourse = new PartTimeCourse("Python", 5, 1500, "Morning");

```

```

            // Creating students

```

```

            Student student1 = new Student(1, "Alish", csharpCourse);

```

```

            Student student2 = new Student(2, "Alishma", aspNetCourse);

```

```

            Student student3 = new Student(3, "Karishma", pythonCourse);

```

```

            // Simulate payments flow

```

```
student1.Payment(500);  
student2.Payment(1000);  
student3.Payment(1500);
```

```
// Printing student details
```

```
student1.Print();  
student2.Print();  
student3.Print();
```

```
// Printing course details
```

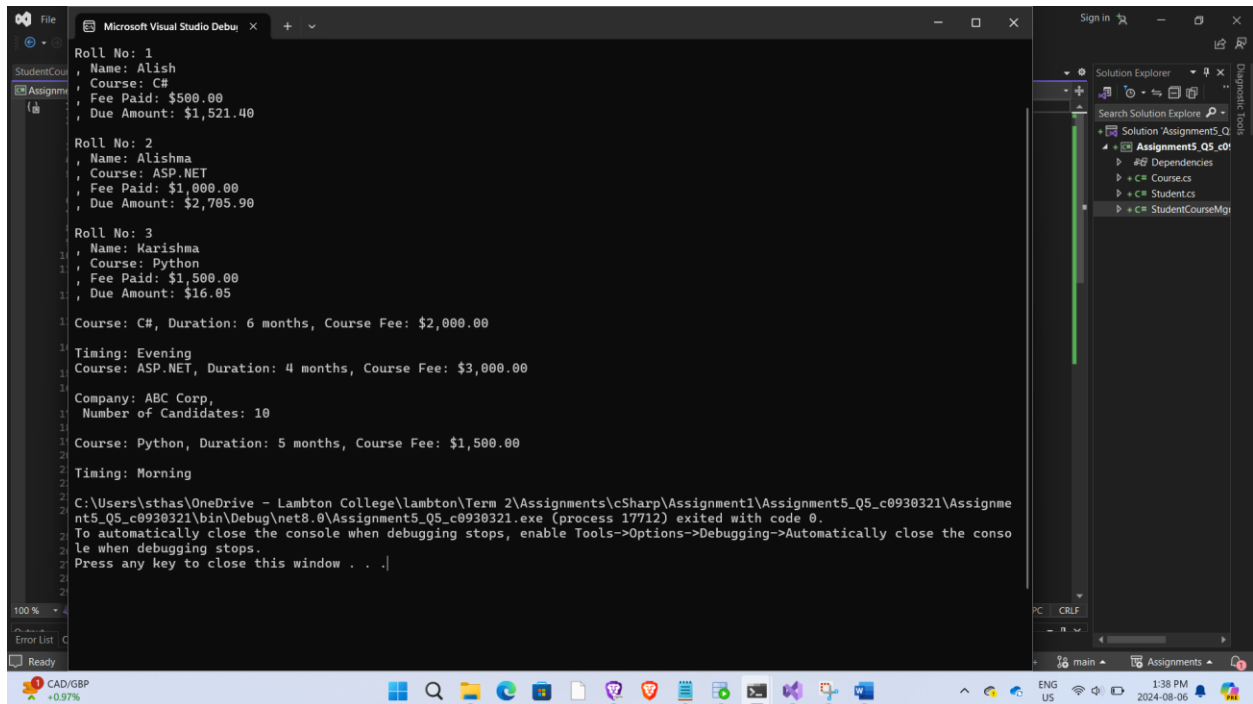
```
csharpCourse.Print();  
aspNetCourse.Print();  
pythonCourse.Print();
```

```
}
```

```
}
```

```
}
```

Output Screenshot:



The screenshot displays the Microsoft Visual Studio IDE in a dark theme. The main window shows the 'Debug Console' with the following output:

```
Roll No: 1
, Name: Alisha
, Course: C#
, Fee Paid: $500.00
, Due Amount: $1,521.40

Roll No: 2
, Name: Alishma
, Course: ASP.NET
, Fee Paid: $1,000.00
, Due Amount: $2,705.90

Roll No: 3
, Name: Marishma
, Course: Python
, Fee Paid: $1,500.00
, Due Amount: $16.05

Course: C#, Duration: 6 months, Course Fee: $2,000.00

Timing: Evening
Course: ASP.NET, Duration: 4 months, Course Fee: $3,000.00

Company: ABC Corp,
Number of Candidates: 10

Course: Python, Duration: 5 months, Course Fee: $1,500.00

Timing: Morning

C:\Users\sthas\OneDrive - Lambton College\lambton\Term 2\Assignments\cSharp\Assignment1\Assignment5_Q5_c0930321\Assignme
nt5_Q5_c0930321\bin\Debug\net8.0\Assignment5_Q5_c0930321.exe (process 17712) 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 . . .
```

The right-hand side of the IDE shows the 'Solution Explorer' with a project named 'Assignment5_Q5_c0930321'. The project structure includes:

- Dependencies
- Course.cs
- Student.cs
- StudentCourseMgr

The bottom status bar indicates the system is ready, with a battery level of 100% and a network connection to 'main'.