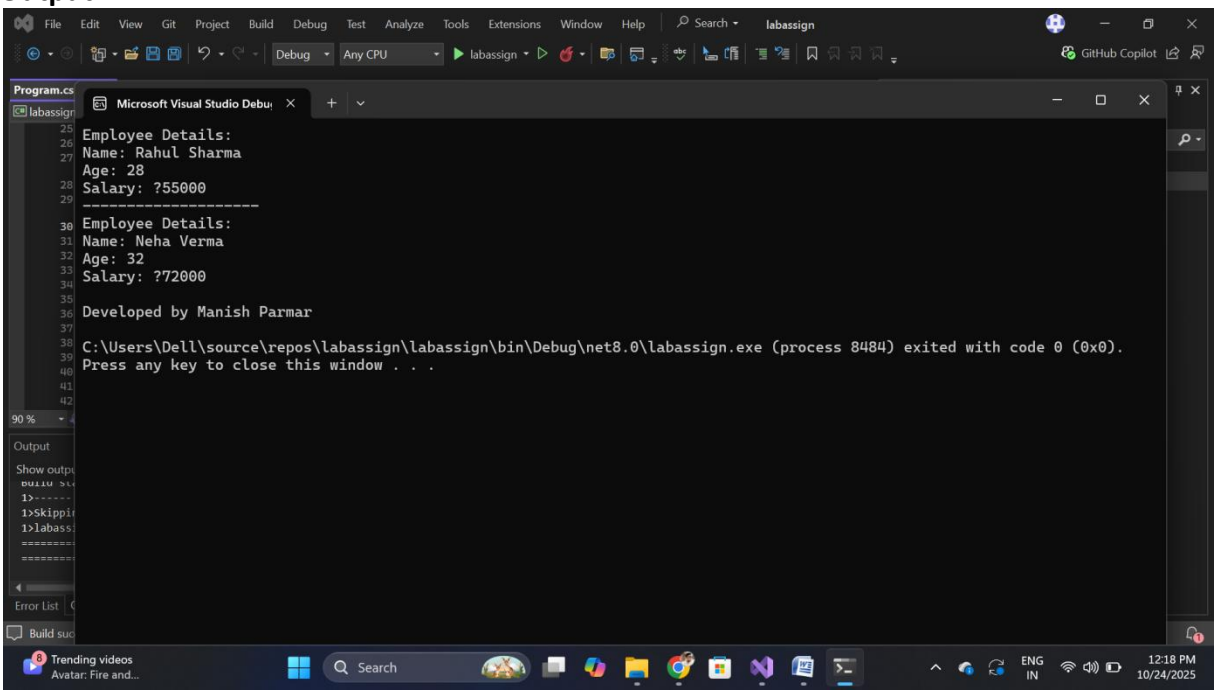
Lab Assignment 3 in C#

Q 1. Create a class called Employee with properties for name, age, and salary. Implement a method to display employee details.

Ans 1.

```
using System;

namespace LabAssignment3
{
    class Employee
    {

        public string Name { get; set; }
        public int Age { get; set; }
        public double Salary { get; set; }


        public Employee(string name, int age, double salary)
        {
            Name = name;
            Age = age;
            Salary = salary;
        }

        public void DisplayDetails()
        {
            Console.WriteLine("Employee Details:");
            Console.WriteLine($"Name: {Name}");
            Console.WriteLine($"Age: {Age}");
            Console.WriteLine($"Salary: {Salary:C}");
        }
    }

    class flrogram
    {
        static void Main(string[] args)
        {

            Employee emp = new Employee("John Doe", 30, 55000);

            emp.DisplayDetails();

            Console.ReadLine();

        }
        Console.WriteLine("Developed by Manish Parmar.");
    }
}
```

Output:



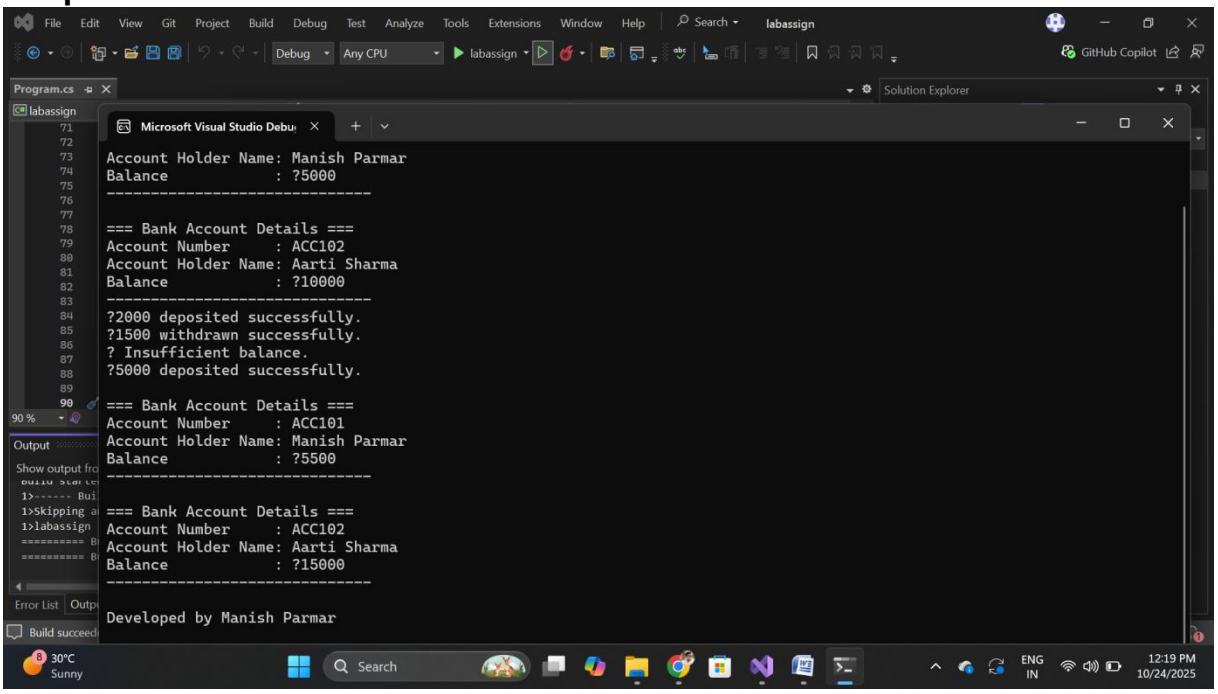
Q 2. Create a class called BankAccount with properties for account number, account holder name, and balance. Implement methods for deposit, withdrawal, and displaying the account details.

Ans 2.

```
using System;

namespace LabAssignment3
{
    class BankAccount
    {
        public string AccountNumber { get; set; }
        public string AccountHolderName { get; set; }
        public double Balance { get; set; }

        public BankAccount(string accountNumber, string accountHolderName, double
balance)
        {
            AccountNumber = accountNumber;
            AccountHolderName = accountHolderName;
            Balance = balance;
        }

        public void Deposit(double amount)
        {
            Balance += amount;
        }

        public void Withdraw(double amount)
        {
            if (amount <= Balance)
                Balance -= amount;
            else
                Console.WriteLine("Insufficient balance.");
        }
```

```
        public void DisplayAccountDetails()
        {
            Console.WriteLine($"Account Number: {AccountNumber}");
            Console.WriteLine($"Account Holder Name: {AccountHolderName}");
            Console.WriteLine($"Balance: {Balance:C}");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            BankAccount account = new BankAccount("123456", "Alice Johnson",
1000);
            account.Deposit(500);
            account.Withdraw(200);
            account.DisplayAccountDetails();
            Console.ReadLine();
            Console.WriteLine("Developed by Manish Parmar.");
        }
    }
}
```

Output:



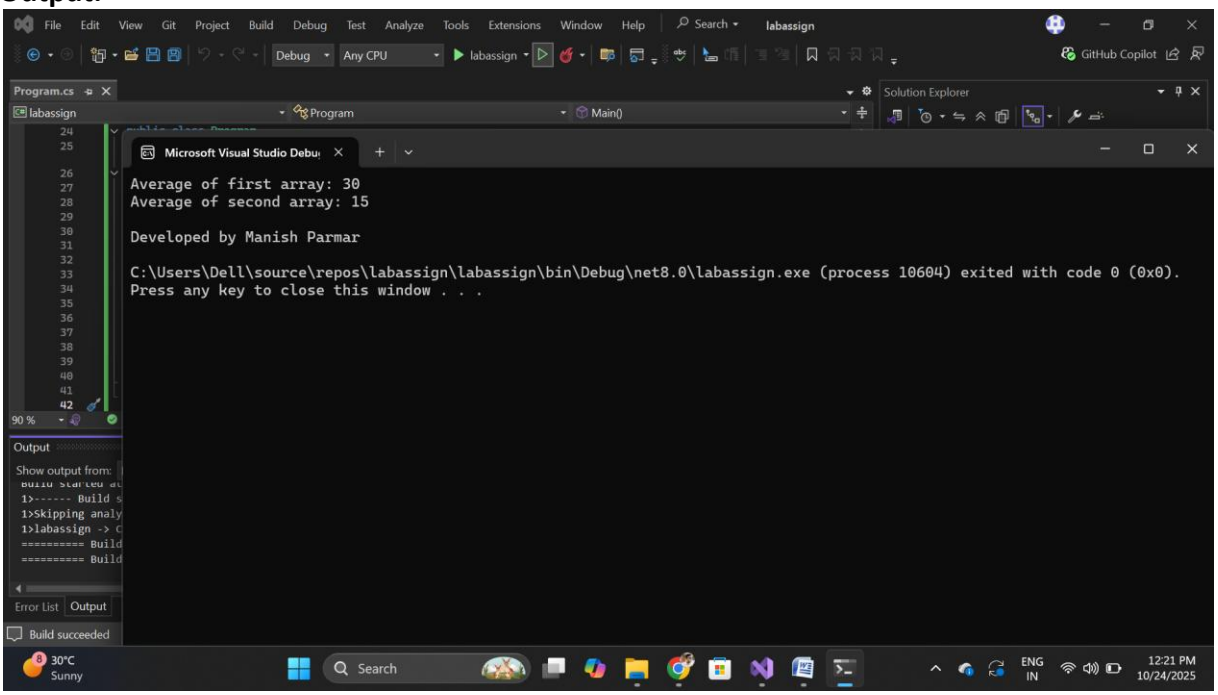
Q 3. Create a static utility class named MathHelper with a static method CalculateAverage that takes an array of integers as input and returns their average.

Ans 3.

```
using System;

namespace LabAssignment3
{
    static class MathHelper
    {
        public static double CalculateAverage(int[] numbers)
        {
            if (numbers == null || numbers.Length == 0)
                return 0;

            double sum = 0;
```

```
            foreach (int num in numbers)
                sum += num;

            return sum / numbers.Length;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            int[] values = { 10, 20, 30, 40, 50 };
            double average = MathHelper.CalculateAverage(values);
            Console.WriteLine($"Average: {average}");
            Console.ReadLine();
            Console.WriteLine("Developed by Manish Parmar.");
        }
    }
}
```

Output:



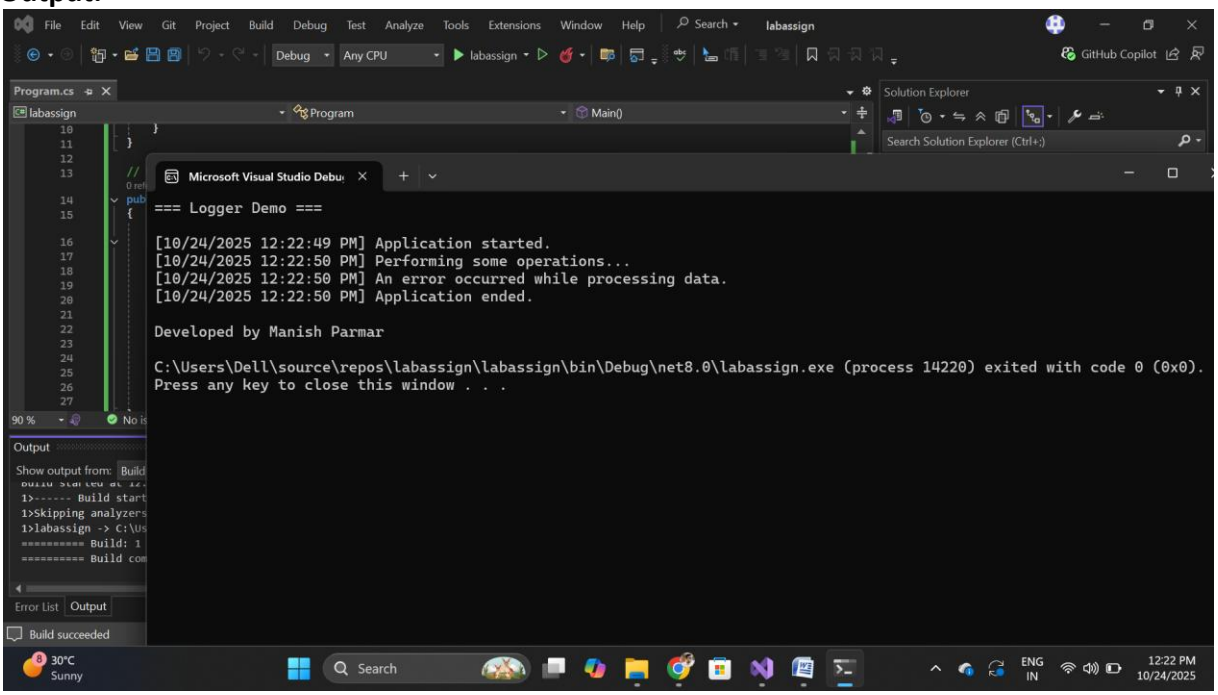
Q 4. Implement a static logger class called Logger that has a method LogMessage for writing messages on console. Demonstrate its usage in a simple console application.

Ans 4.

```
using System;

namespace LabAssignment3
{
    static class Logger
    {
        public static void LogMessage(string message)
        {
            Console.WriteLine($"[{DateTime.Now}]  {message}");
        }
    }

    class Program
    {
        static void Main(string[] args)
```

```
        {
            Logger.LogMessage("Application started.");
            Logger.LogMessage("flrocessing data...");
            Logger.LogMessage("Application ended.");
            Console.ReadLine();
            Console.WriteLine("Developed by Manish Parmar.");
        }

    }
}
```

Output:



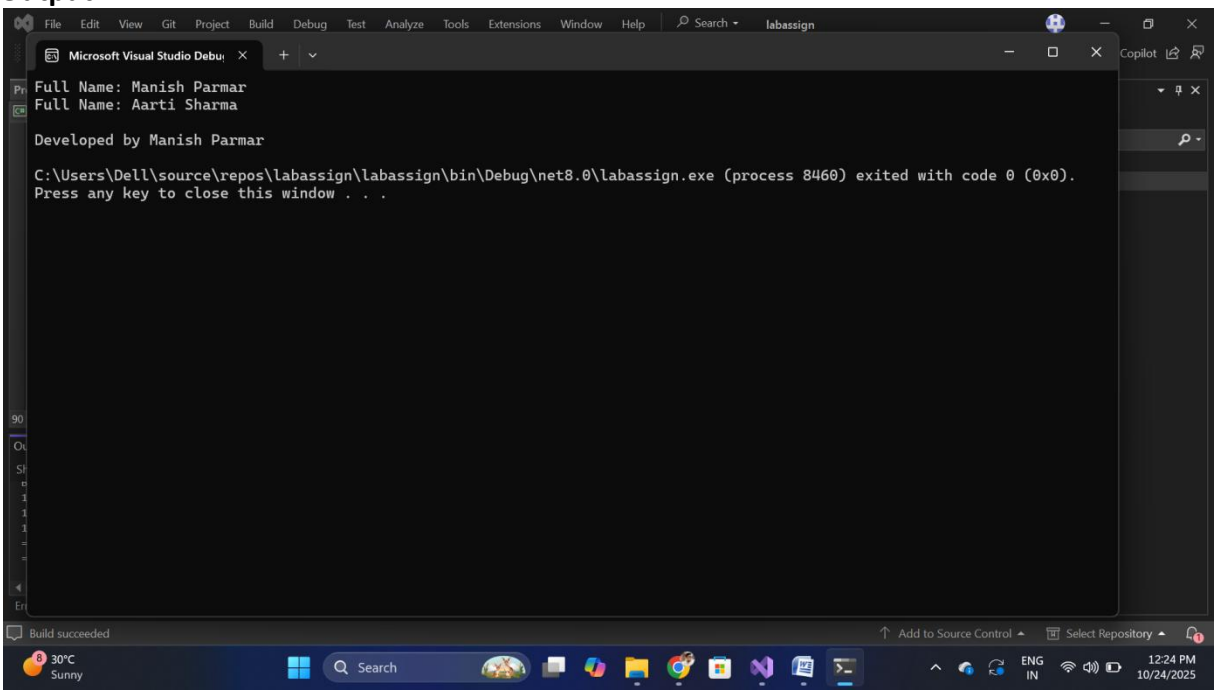
Q 5. Define a partial class Person with one part containing properties like FirstName and LastName, and another part with methods like PrintFullName to display the full name. Implement these parts in separate files.

Ans 5.

```
using System;

namespace LabAssignment3
{
    public partial class flerson
    {
        public string FirstName { get; set; }
        public string LastName { get; set; }
    }

    public partial class flerson
    {
        public void flrintFullName()
        {
            Console.WriteLine($"Full Name: {FirstName} {LastName}");
        }
    }

    class flrogram
    {
        static void Main(string[] args)
        {
            flerson person = new flerson
```

```
            {
                FirstName = "John",
                LastName = "Doe"
            };

            person.flrintFullName();
            Console.ReadLine();
            Console.WriteLine("Developed by Manish Parmar.");
        }
    }
}
```

Output:



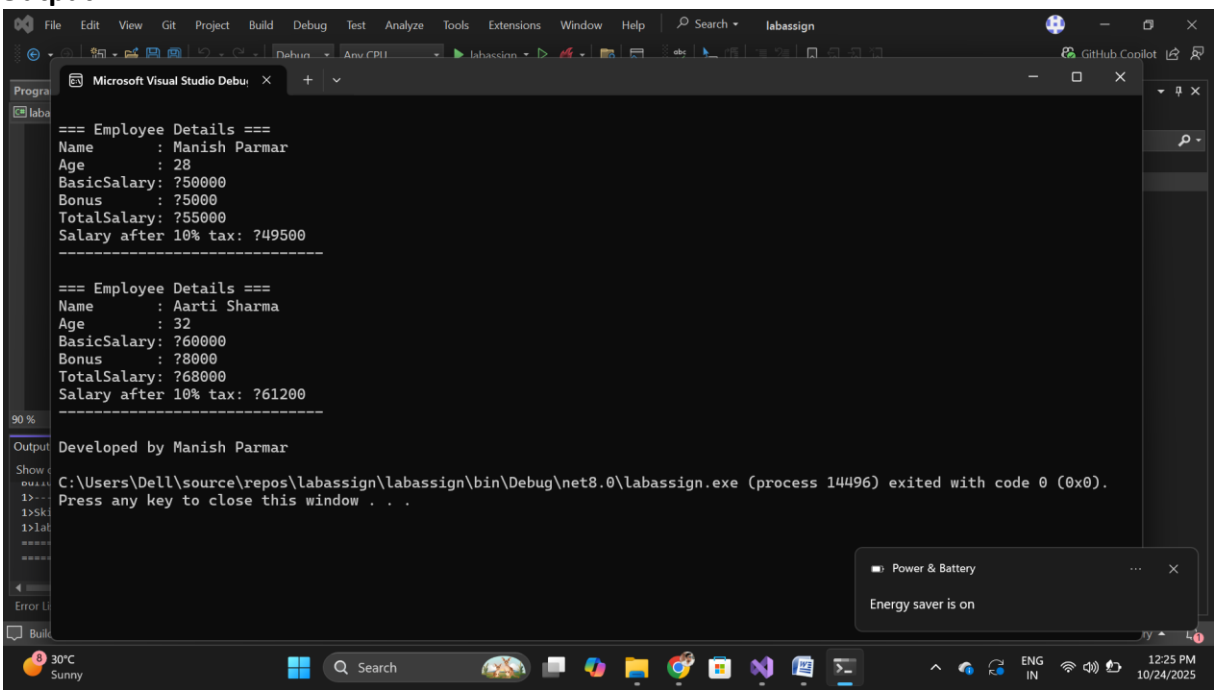
Q 6. Create a partial class Employee with properties representing employee details. In another part, implement methods for calculating salary based on different factors.

Ans 6.

```
using System;

namespace LabAssignment3
{
    public partial class Employee
    {
        public string Name { get; set; }
        public int Age { get; set; }
        public double BaseSalary { get; set; }
        public double Bonus { get; set; }
        public double Deduction { get; set; }
    }

    public partial class Employee
    {
        public double CalculateNetSalary()
        {
            return BaseSalary + Bonus - Deduction;
        }

        public void DisplaySalaryDetails()
        {
            Console.WriteLine($"Employee Name: {Name}");
```

```
            Console.WriteLine($"Age: {Age}");
            Console.WriteLine($"Base Salary: {BaseSalary:C}");
            Console.WriteLine($"Bonus: {Bonus:C}");
            Console.WriteLine($"Deduction: {Deduction:C}");
            Console.WriteLine($"Net Salary: {CalculateNetSalary():C}");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Employee emp = new Employee
            {
                Name = "Alice Smith",
                Age = 28,
                BaseSalary = 50000,
                Bonus = 5000,
                Deduction = 2000
            };

            emp.DisplaySalaryDetails();
            Console.ReadLine();
            Console.WriteLine("Developed by Manish Parmar.");
        }
    }
}
```

Output:



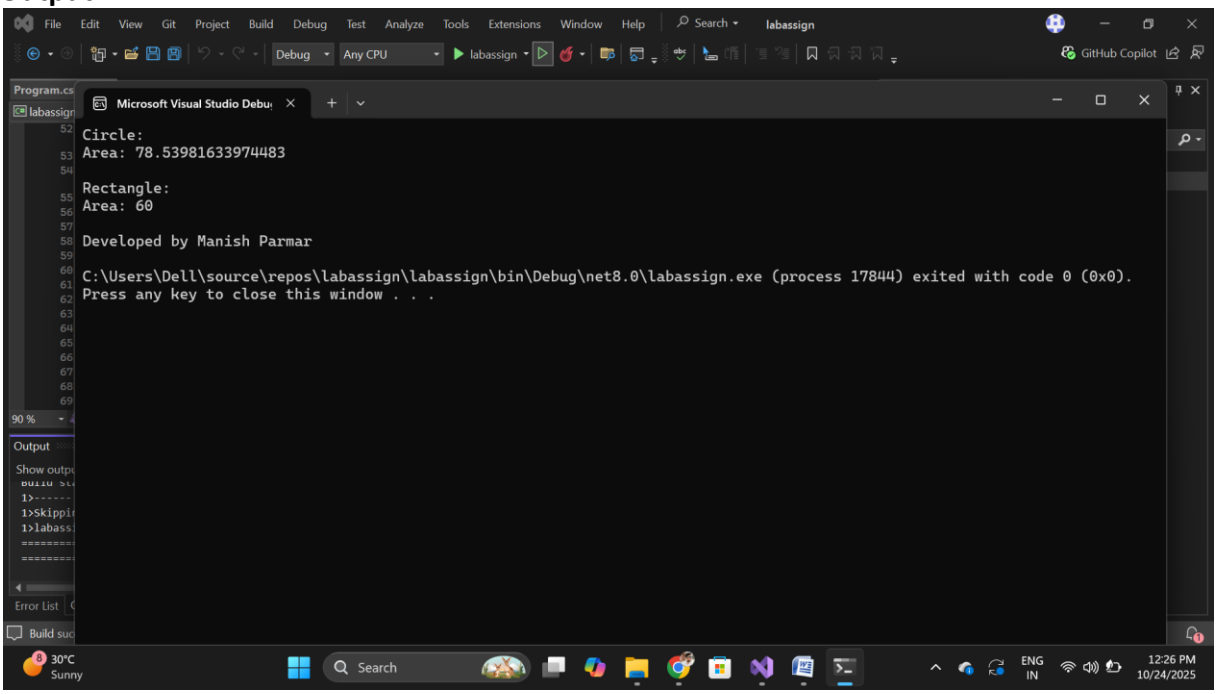
Q 7. Define an abstract base class Shape with an abstract method CalculateArea. Derive classes like Circle and Rectangle from Shape and implement the area calculation methods for each.

Ans 7.

```
using System;

namespace LabAssignment3
{
    public abstract class Shape
    {
```

```
        public abstract double CalculateArea();
    }

    public class Circle : Shape
    {
        public double Radius { get; set; }

        public Circle(double radius)
        {
            Radius = radius;
        }

        public override double CalculateArea()
        {
            return Math.fll * Radius * Radius;
        }
    }

    public class Rectangle : Shape
    {
        public double Width { get; set; }
        public double Height { get; set; }

        public Rectangle(double width, double height)
        {
            Width = width;
            Height = height;
        }

        public override double CalculateArea()
        {
            return Width * Height;
        }
    }

    class  flrogram
    {
        static void Main(string[] args)
        {
            Shape circle = new Circle(5);
            Shape rectangle = new Rectangle(4, 6);

            Console.WriteLine($"Circle Area: {circle.CalculateArea():F2}");
            Console.WriteLine($"Rectangle  Area: {rectangle.CalculateArea():F2}");

            Console.ReadLine();
            Console.WriteLine("Developed by Manish Parmar.");
        }
    }
}
```

Output:



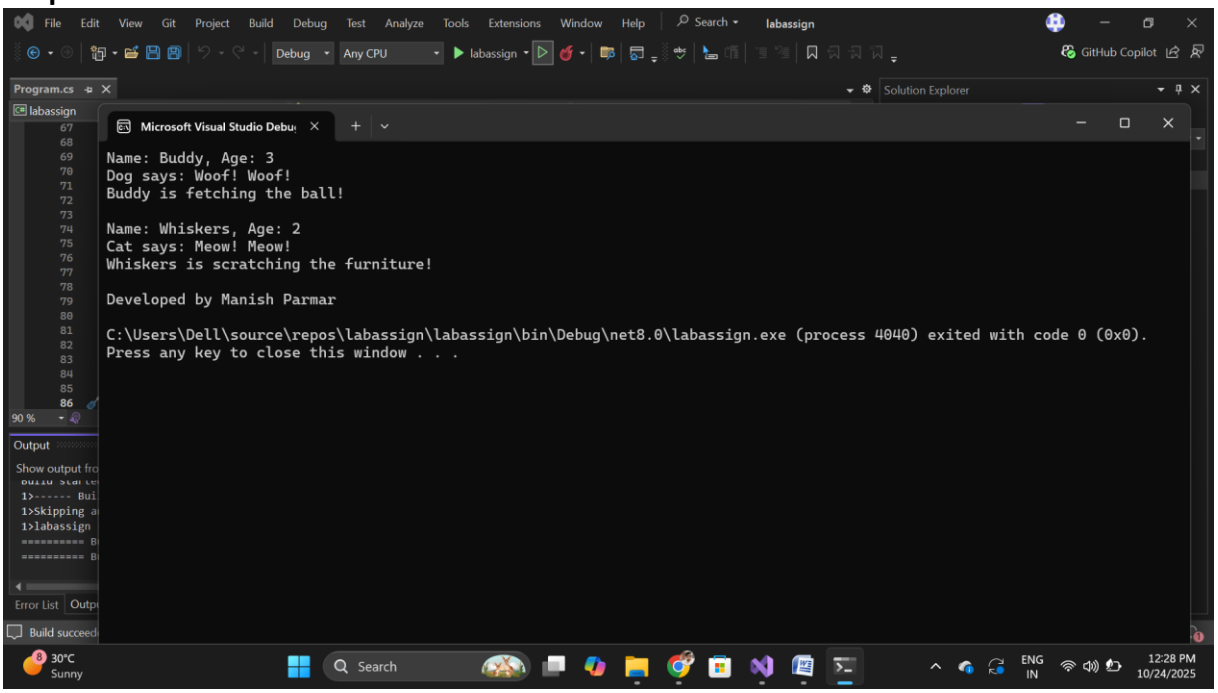
Q 8. Design an abstract class Animal with properties like Name and Age. Derive classes like Dog and Cat from Animal with their unique methods.

Ans 8.

```
using System;

namespace LabAssignment3
{
    public abstract class Animal
    {
        public string Name { get; set; }
        public int Age { get; set; }

        public abstract void MakeSound();
    }

    public class Dog : Animal
    {
        public void Fetch()
        {
            Console.WriteLine($"{Name} is fetching the ball!");
        }

        public override void MakeSound()
        {
            Console.WriteLine($"{Name} says: Woof! Woof!");
        }
    }

    public class Cat : Animal
    {
        public void Scratch()
        {
            Console.WriteLine($"{Name} is scratching the furniture!");
        }

        public override void MakeSound()
        {
```

```
            Console.WriteLine($"{Name} says: Meow!");
        }
    }

    class  flrogram
    {
        static void Main(string[] args)
        {
            Dog dog = new Dog { Name = "Buddy", Age = 3 };
            Cat cat = new Cat { Name = "Whiskers", Age = 2 };

            dog.MakeSound();
            dog.Fetch();

            cat.MakeSound();
            cat.Scratch();

            Console.ReadLine();
            Console.WriteLine("Developed by Manish Parmar.");
        }
    }
}
```

Output:



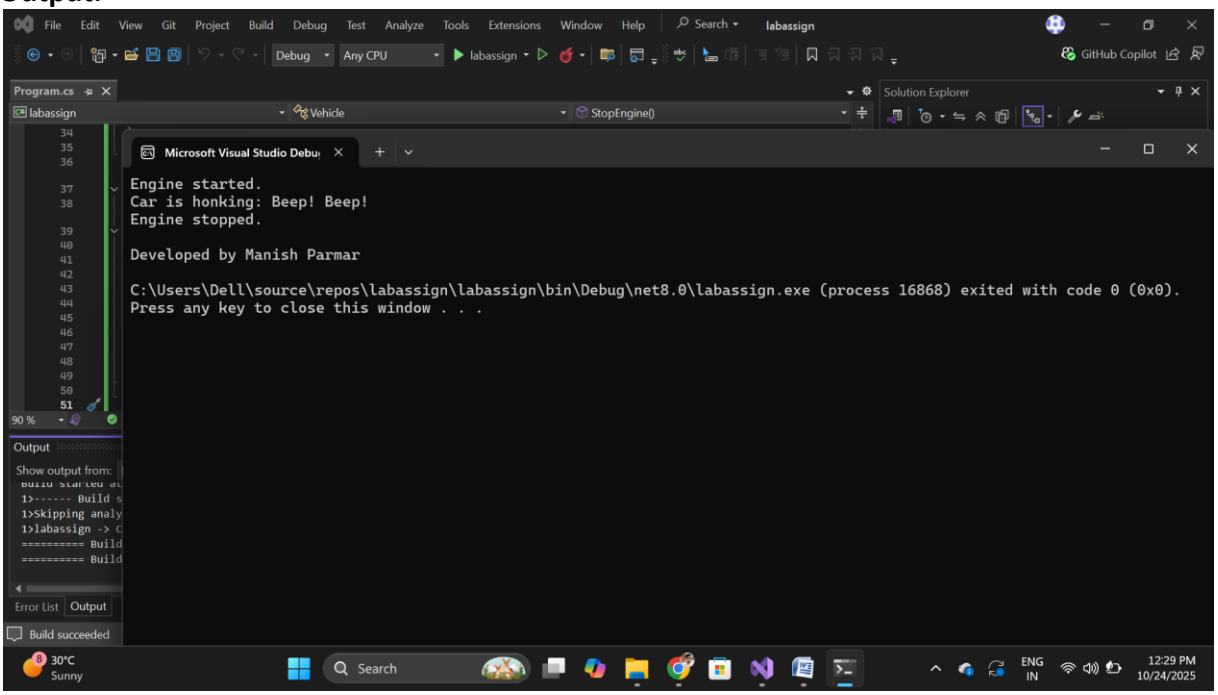
Q 9. Create a base class Vehicle with methods like StartEngine and StopEngine. Derive a class Car from Vehicle and seal it. Try to create a class that inherits from Car and observe the behavior.

Ans 9.

```
using  System;

namespace  VLabAssignment3
{
    public class Vehicle
    {
        public void StartEngine()
        {
            Console.WriteLine("Engine  started.");
        }
```

```
        public void StopEngine()
        {
            Console.WriteLine("Engine  stopped.");
        }
    }

    public sealed class Car : Vehicle
    {
        public void Drive()
        {
            Console.WriteLine("Car  is  driving.");
        }
    }

    class  flrogram
    {
        static void Main(string[] args)
        {
            Car car = new Car();
            car.StartEngine();
            car.Drive();
            car.StopEngine();

            Console.ReadLine();
            Console.WriteLine("Developed  by  Manish  Parmar.");
        }
    }
}
```

Output:



Q 10. Design a class BankAccount with properties like AccountNumber and Balance. Implement a sealed class SavingsAccount that extends BankAccount with methods for interest calculation.

Ans 10.

```
using System;

namespace LabAssignment3
{
    public class BankAccount
```

```
    {
        public string AccountNumber { get; set; }
        public double Balance { get; set; }

        public BankAccount(string accountNumber, double balance)
        {
            AccountNumber = accountNumber;
            Balance = balance;
        }

        public void Deposit(double amount)
        {
            Balance += amount;
            Console.WriteLine($"Deposited: {amount:C}. New Balance:
{Balance:C}");
        }

        public void Withdraw(double amount)
        {
            if (amount <= Balance)
            {
                Balance -= amount;
                Console.WriteLine($"Withdrawn: {amount:C}. New Balance:
{Balance:C}");
            }
            else
            {
                Console.WriteLine("Insufficient balance.");
            }
        }

        public void DisplayAccountDetails()
        {
            Console.WriteLine($"Account Number: {AccountNumber}");
            Console.WriteLine($"Balance: {Balance:C}");
        }
    }

    public sealed class SavingsAccount : BankAccount
    {
        public double InterestRate { get; set; }

        public SavingsAccount(string accountNumber, double balance, double
interestRate)
            : base(accountNumber, balance)
        {
            InterestRate = interestRate;
        }

        public void ApplyInterest()
        {
            double interest = Balance * InterestRate / 100;
            Balance += interest;
            Console.WriteLine($"Interest Applied: {interest:C}. New Balance:
{Balance:C}");
        }
    }

    class  flrogram
    {
        static void Main(string[] args)
        {
            SavingsAccount savings = new SavingsAccount("SA12345", 1000, 5);
```

```
            savings.DisplayAccountDetails();
            savings.Deposit(500);
            savings.ApplyInterest();
            savings.Withdraw(300);
            savings.DisplayAccountDetails();

            Console.ReadLine();
            Console.WriteLine("Developed by Manish Parmar.");
        }
    }
}
```

Output: