International University of Business Agriculture and Technology

Department: Computer Science and Engineering

Semester: Spring 2025

Course Name: Visual Programming

Course Code: CSC 440

Section: A

Lab Report topic: Lab task 03

Submitted To:

Suhala Lamia
Assistant Professor
Department of Computer Science and Engineering

Submitted By:

Samiul Karim Mazumder
22303308

Date of Submission: 21/03/25

Experiment No. 03: Inheritance hierarchy in C#

Objective: The objective of this lab is to understand and implement inheritance in C# by creating class hierarchies that demonstrate different real-life scenarios. This includes:

- Implementing base and derived classes.
- Using method overriding for specialized behavior.
- Utilizing properties to encapsulate data.

Algorithm:

3.1. Vehicle Class Hierarchy

1. Define a base class Vehicle with attributes:
   - Model (string)
   - Year (integer)
   - FuelType (string)
   - Virtual methods: DisplayInfo(), FuelEfficiency(), and MaxSpeed()
2. Create three derived classes: Car, Truck, and Motorcycle, inheriting from the class Vehicle.
3. Override the FuelEfficiency() and MaxSpeed() methods for each subclass with appropriate values.
4. Implement the RunVehicleProgram() method to instantiate and display information for a Car, Truck, and Motorcycle.
5. Call DisplayInfo(), FuelEfficiency(), and MaxSpeed() for each object.

3.2. Shape Class Hierarchy

1. Define a base class Shape with a virtual method CalculateArea().
2. Create three subclasses: Circle, Rectangle, and Triangle, inheriting from Shape.
3. Override CalculateArea() in each subclass to compute the area based on:
   - Circle: π * radius^2
   - Rectangle: length * width
   - Triangle: 0.5 * baseLength * height
4. Implement RunShapeProgram() to:
   - Take user input for shape dimensions.
   - Instantiate the corresponding shape object.
   - Display the computed area.

3.3. Multilevel Inheritance

1. Create a base class Inherit with:
    - A string attribute str.
    - A method InheritMethod() to display the attribute.
2. Create a subclass InheritChild that inherits from Inherit and defines InheritChildMethod().
3. Create another subclass Child, inheriting from InheritChild, and define ShowAll() to call all parent class methods.
4. Implement RunInheritanceProgram() to instantiate a Child object and invoke ShowAll().

3.4. Animal Class Hierarchy

1. Define a base class Animal with methods Walk() and Eat().
2. Create two subclasses:
    - Dog with an additional method Bark().
    - Bird with an additional method Fly().
3. Create a subclass Dove, inheriting from Bird, with an attribute Color and a method ShowColor().
4. Implement RunAnimalProgram() to instantiate a Dove object and invoke its methods.

Program:

Main Program:

```
using System;

class Program
{
   static void Main()
   {
      Console.WriteLine("Select a program to run:");
      Console.WriteLine("1. Vehicle Class Hierarchy");
      Console.WriteLine("2. Shape Class Hierarchy");
      Console.WriteLine("3. Multilevel Inheritance");
      Console.WriteLine("4. Animal Class Hierarchy");
      Console.Write("Enter choice: ");
      int choice = int.Parse(Console.ReadLine());

      switch (choice)
```

```
    {
        case 1:
            RunVehicleProgram();
            break;
        case 2:
            RunShapeProgram();
            break;
        case 3:
            RunInheritanceProgram();
            break;
        case 4:
            RunAnimalProgram();
            break;
        default:
            Console.WriteLine("Invalid choice");
            break;
    }
}

static void RunVehicleProgram()
{
    Vehicle car = new Car("Toyota", 2022, "Petrol");
    Vehicle truck = new Truck("Ford", 2020, "Diesel");
    Vehicle motorcycle = new Motorcycle("Honda", 2021, "Petrol");

    car.DisplayInfo();
    Console.WriteLine("Fuel Efficiency: " + car.FuelEfficiency() + " km/l");
    Console.WriteLine("Max Speed: " + car.MaxSpeed() + " km/h");
    Console.WriteLine();

    truck.DisplayInfo();
    Console.WriteLine("Fuel Efficiency: " + truck.FuelEfficiency() + " km/l");
    Console.WriteLine("Max Speed: " + truck.MaxSpeed() + " km/h");
    Console.WriteLine();

    motorcycle.DisplayInfo();
    Console.WriteLine("Fuel Efficiency: " + motorcycle.FuelEfficiency() + " km/l");
```

```
  Console.WriteLine("Max Speed: " + motorcycle.MaxSpeed() + " km/h");
}

static void RunShapeProgram()
{
  Console.WriteLine("Enter the radius of the circle:");
  int radius = int.Parse(Console.ReadLine());
  Circle circle = new Circle(radius);
  Console.WriteLine("Area of the circle: " + circle.CalculateArea());

  Console.Write("Enter the length of the rectangle: ");
  double length = double.Parse(Console.ReadLine());
  Console.Write("Enter the width of the rectangle: ");
  double width = double.Parse(Console.ReadLine());
  Rectangle rectangle = new Rectangle(length, width);
  Console.WriteLine("Area of the rectangle: " + rectangle.CalculateArea());

  Console.Write("Enter the base length of the triangle: ");
  double baseLength = double.Parse(Console.ReadLine());
  Console.Write("Enter the height of the triangle: ");
  double height = double.Parse(Console.ReadLine());
  Triangle triangle = new Triangle(baseLength, height);
  Console.WriteLine("Area of the triangle: " + triangle.CalculateArea());

}

static void RunInheritanceProgram()
{
  Child obj = new Child();
  obj.ShowAll();
}

static void RunAnimalProgram()
{
  Dove dove = new Dove();
  dove.Walk();
  dove.Fly();
```

```
        dove.ShowColor();
    }
}
```

Vehicle.cs:

```
class Vehicle
{
    public string Model;
    public int Year;
    public string FuelType;

    public Vehicle(string model, int year, string fuelType)
    {
        Model = model;
        Year = year;
        FuelType = fuelType;
    }

    public virtual void DisplayInfo()
    {
        Console.WriteLine($"Model: {Model}, Year: {Year}, Fuel: {FuelType}");
    }

    public virtual double FuelEfficiency()
    {
        return 0;
    }

    public virtual double MaxSpeed()
    {
        return 0;
    }
}

class Car : Vehicle
{
    public Car(string model, int year, string fuelType) : base(model, year, fuelType) { }
    public override double FuelEfficiency()
    {
        return 15;
    }

    public override double MaxSpeed()
    {
        return 180;
    }
}

class Truck : Vehicle
{
    public Truck(string model, int year, string fuelType) : base(model, year, fuelType) { }

    public override double FuelEfficiency()
    {
        return 8;
    }

    public override double MaxSpeed()
    {
        return 120;
    }
}

class Motorcycle : Vehicle
{
    public Motorcycle(string model, int year, string fuelType) : base(model, year,
fuelType) { }

    public override double FuelEfficiency()
    {
        return 40;
    }
    public override double MaxSpeed()
    {
        return 220;
    }
}
```

Shape.cs:

```
using System;
public class Shape
{
    public int r;

    public virtual double CalculateArea()
    {
        return 0;
    }

}

class Circle : Shape
{
    private double radius;

    public Circle(double radius)
    {
        this.radius = radius;
    }
    public override double CalculateArea()
    {
        return 3.1416 * radius * radius;
    }
}

public class Rectangle : Shape
{
    private double length, width;
    public Rectangle(double length, double width)
    {
        this.length = length;
        this.width = width;
    }

    public override double CalculateArea()
    {
        return length * width;
    }
}


public class Triangle : Shape
{
    private double baseLength, height;

    public Triangle(double baseLength, double height)
    {
        this.baseLength = baseLength;
        this.height = height;
    }

    public override double CalculateArea()
    {
        return 0.5 * baseLength * height;
    }
}
```

Inherit.cs:

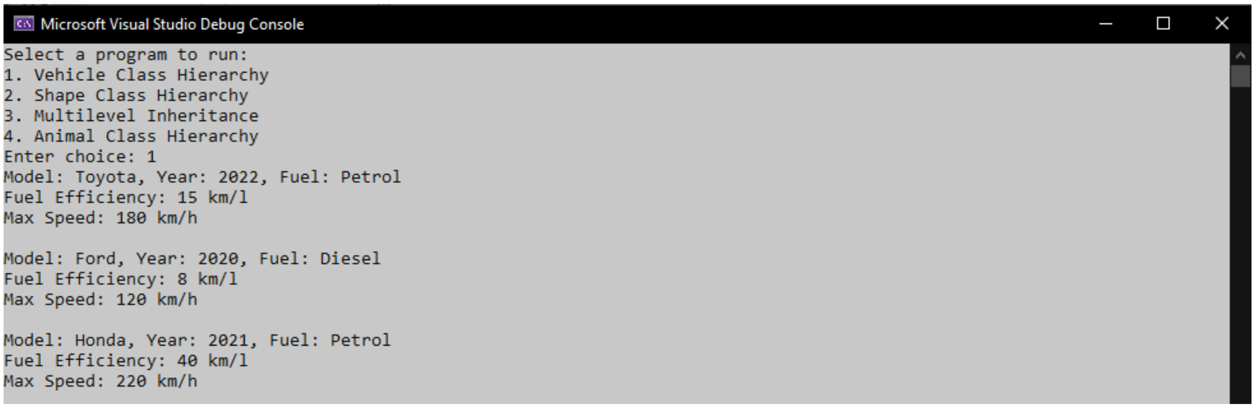
```
using System;
class Inherit
{
    public string str = "Inherit class";
    public void InheritMethod()
    {
        Console.WriteLine("Inherit method: " + str);
    }
}

class InheritChild : Inherit
{
    public void InheritChildMethod()
    {
        Console.WriteLine("InheritChild method called. Accessing parent class variable: " +
str);
    }
}

class Child : InheritChild
{
    public void ShowAll()
    {
        Console.WriteLine("Child method called. Accessing all parent class properties:");
        InheritMethod();
        InheritChildMethod();
    }
}
```

Animal.cs:

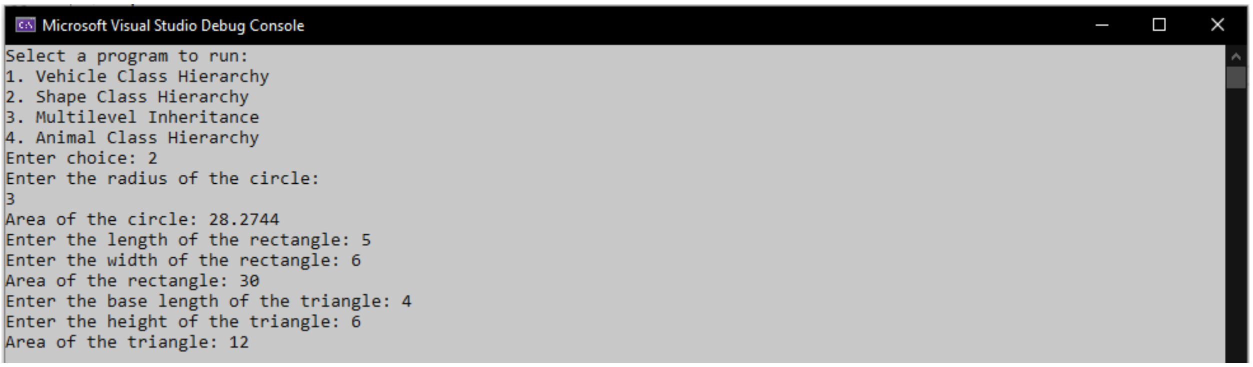
```
using System;

class Animal
{
    public void Walk()
    {
        Console.WriteLine("This animal is walking.");
    }

    public void Eat()
    {
        Console.WriteLine("This animal is eating.");
    }
}
```

```
class Dog : Animal
{
   public int NoOfLegs { get; set; }
   public void Bark()
   {
      Console.WriteLine("The dog is barking.");
   }
}

class Bird : Animal
{
   public int NoOfWings { get; set; }
   public void Fly()
   {
      Console.WriteLine("The bird is flying.");
   }
}

class Dove : Bird
{
   public string Color { get; set; }
   public void ShowColor()
   {
      Console.WriteLine("The dove's color is " + Color + ".");
   }
}
```

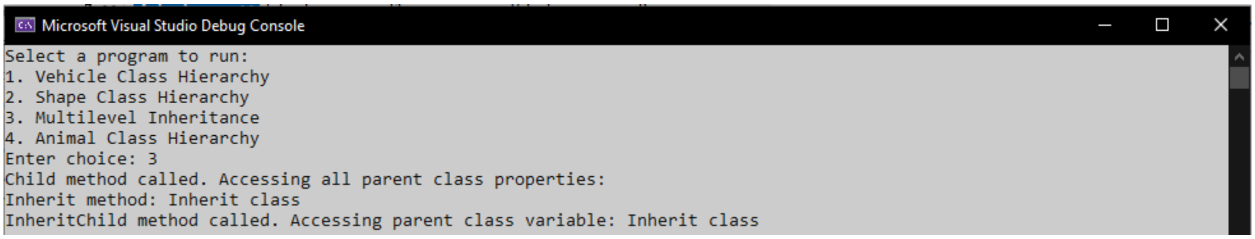
Output:



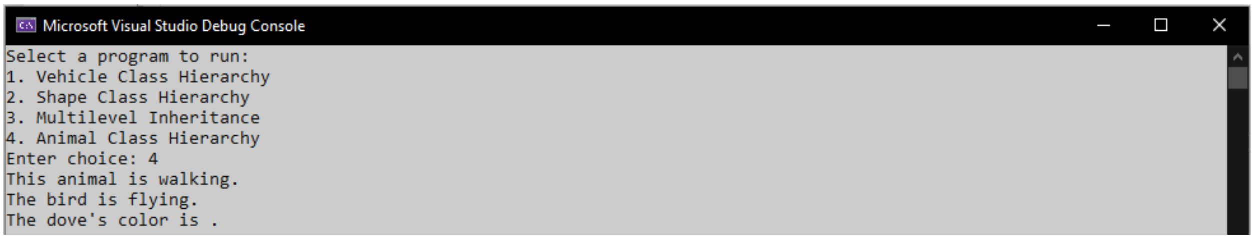
```
Select a program to run:
1. Vehicle Class Hierarchy
2. Shape Class Hierarchy
3. Multilevel Inheritance
4. Animal Class Hierarchy
Enter choice: 1
Model: Toyota, Year: 2022, Fuel: Petrol
Fuel Efficiency: 15 km/l
Max Speed: 180 km/h

Model: Ford, Year: 2020, Fuel: Diesel
Fuel Efficiency: 8 km/l
Max Speed: 120 km/h

Model: Honda, Year: 2021, Fuel: Petrol
Fuel Efficiency: 40 km/l
Max Speed: 220 km/h
```



```
Select a program to run:
1. Vehicle Class Hierarchy
2. Shape Class Hierarchy
3. Multilevel Inheritance
4. Animal Class Hierarchy
Enter choice: 2
Enter the radius of the circle:
3
Area of the circle: 28.2744
Enter the length of the rectangle: 5
Enter the width of the rectangle: 6
Area of the rectangle: 30
Enter the base length of the triangle: 4
Enter the height of the triangle: 6
Area of the triangle: 12
```



```
Select a program to run:
1. Vehicle Class Hierarchy
2. Shape Class Hierarchy
3. Multilevel Inheritance
4. Animal Class Hierarchy
Enter choice: 3
Child method called. Accessing all parent class properties:
Inherit method: Inherit class
InheritChild method called. Accessing parent class variable: Inherit class
```



```
Select a program to run:
1. Vehicle Class Hierarchy
2. Shape Class Hierarchy
3. Multilevel Inheritance
4. Animal Class Hierarchy
Enter choice: 4
This animal is walking.
The bird is flying.
The dove's color is .
```