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 04

Submitted To:

Suhala Lamia
Assistant Professor
Department of Computer Science and Engineering

Submitted By:

Samiul Karim Mazumder
22303308

Date of Submission: 27/03/25

Experiment No. 03: Polymorphism hierarchy in C#

Objective: The objective of this lab is to understand and implement **inheritance and polymorphism** in C#.

Algorithm:

4.1. Employee Salary Calculation

1. **Create a base class Employee with properties Name and Salary and a virtual method CalculateSalary().**

2. **Create two derived classes:**

    - **Manager: Overrides CalculateSalary() to include a bonus calculation.**

    - **Programmer: Overrides CalculateSalary() to be based on projects.**

3. **In the Main() method:**

    - **Create instances of Manager and Programmer.**

    - **Call CalculateSalary() on both to demonstrate polymorphism.**

4.2. Animal Sound Behavior

1. **Create a base class Animal with a virtual method Speak().**

2. **Create three derived classes:**

    - **Dog: Overrides Speak() to print "Dog barks."**

    - **Cow: Overrides Speak() to print "Cow moos."**

    - **Cat: Overrides Speak() to print "Cat meows."**

3. **In the Main() method:**

- **Create instances of Dog, Cow, and Cat.**
- **Call Speak() on each object to demonstrate polymorphism.**

Program:

Program.cs:

```
using System;

namespace lab4
{
   public class Program
   {
      static void Main()
      {
         Console.WriteLine("Employee Salary Calculation:");
         Employee manager = new Manager("Alice", 45000);
         Employee programmer = new Programmer("Bob", 5000);
         manager.CalculateSalary();
         programmer.CalculateSalary();

         Console.WriteLine("\nAnimal Sounds:");
         Animal dog = new Dog();
         Animal cow = new Cow();
         Animal cat = new Cat();
         dog.Speak();
         cow.Speak();
         cat.Speak();
      }
   }
}
```

Employee.cs:

```
using System;
```

```
namespace lab4
{
    class Employee
    {
        public string Name;
        public int Salary;

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

        public virtual void CalculateSalary()
        {
            Console.WriteLine("Salary calculation is based on role.");
        }
    }

    class Manager : Employee
    {
        public Manager(string name, int salary) : base(name, salary) { }

        public override void CalculateSalary()
        {
            Console.WriteLine($"Manager {Name}'s salary is calculated with bonuses:
{Salary}.");
        }
    }

    class Programmer : Employee
    {
        public Programmer(string name, int salary) : base(name, salary) { }

        public override void CalculateSalary()
        {
```

```
            Console.WriteLine($"Programmer {Name}'s salary is calculated based on
projects: {Salary}.");
        }
    }
}
```

Animal.cs:

```
using System;

namespace lab4
{
    class Animal
    {
        public virtual void Speak()
        {
            Console.WriteLine("Animals make sounds.");
        }
    }

    class Dog : Animal
    {
        public override void Speak()
        {
            Console.WriteLine("Dog barks.");
        }
    }

    class Cow : Animal
    {
        public override void Speak()
        {
            Console.WriteLine("Cow moos.");
        }
    }

    class Cat : Animal
```

```
    {
        public override void Speak()
        {
            Console.WriteLine("Cat meows.");
        }
    }
}
```

Output: