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 05

Submitted To:

Suhala Lamia
Assistant Professor
Department of Computer Science and Engineering

Submitted By:

Samiul Karim Mazumder
22303308

Date of Submission: 8/04/25

Experiment No. 05: Understanding Interfaces and Polymorphism in C#.

Objective: To understand and implement interfaces and polymorphism in C# by solving real-world problems:

- E-commerce payment system using different payment methods
- Smart home system controlling various devices
- Payroll system for processing payments of different types of employees

Algorithm:

➢ **Part 1:** E-Commerce Payment System

- Define a PaymentMethod interface with a method void Pay(double amount).
- Create classes CreditCardPayment, PayPalPayment, and BankTransferPayment implementing the interface.
- Each class should override the Pay method with a custom message.
- Create PaymentProcessor class that accepts a PaymentMethod reference and calls Pay().
- In the Main() method, demonstrate dynamic method dispatch.

➢ Part 2: Smart Home Device Control

- Define a SmartDevice interface with methods TurnOn() and TurnOff().
- Implement the interface in Light, Fan, and AC classes.
- Create a SmartHomeController class that can control any device via interface reference.
- In Main(), control different devices using the controller class.

➢ Part 3: Payroll System

- Create a Payable interface with method double CalculatePay().
- Implement FullTimeEmployee, PartTimeEmployee, and Contractor classes.
- Each class overrides CalculatePay() with appropriate logic.
- Create PayrollSystem class that processes a list of Payable employees and prints payment.
- In Main(), create employee instances and display their pay using the system.

Code:

1. **Payment.cs:**

```
using System;
using System.Collections.Generic;

interface Payable
{
    double CalculatePay();
}

class FullTimeEmployee : Payable
{
    private double salary;

    public FullTimeEmployee(double salary)
    {
        this.salary = salary;
    }

    public double CalculatePay()
    {
        return salary;
    }
}

class PartTimeEmployee : Payable
{
    private double hourlyRate;
    private int hoursWorked;

    public PartTimeEmployee(double rate, int hours)
    {
        hourlyRate = rate;
        hoursWorked = hours;
    }

    public double CalculatePay()
    {
        return hourlyRate * hoursWorked;
    }
}

class Contractor : Payable
{
    private double projectPay;

    public Contractor(double pay)
    {
        projectPay = pay;
    }

    public double CalculatePay()
    {
        return projectPay;
    }
}

class PayrollSystem
{
    public void Process(List<Payable> employees)
    {
        foreach (var emp in employees)
        {
            Console.WriteLine($"Payment: ${emp.CalculatePay()}");
        }
    }
}
```

2. **PaymentMethod.cs:**

```
using System;

interface PaymentMethod
{
    void Pay(double amount);
}

class CreditCardPayment : PaymentMethod
{
    public void Pay(double amount)

    {

        Console.WriteLine($"Paid {amount} using Credit Card.");

    }

}


class PayPalPayment : PaymentMethod

{

    public void Pay(double amount)

    {

        Console.WriteLine($"Paid {amount} using PayPal.");

    }

}


class BankTransferPayment : PaymentMethod

{

    public void Pay(double amount)

    {

        Console.WriteLine($"Paid {amount} using Bank Transfer.");

    }

}


class PaymentProcessor

{

    public void Process(PaymentMethod method, double amount)
    {
        method.Pay(amount);
    }
}
```

3. **SmartDevices.cs:**

```
using System;

interface SmartDevice
{
    void TurnOn();
    void TurnOff();
}

class Light : SmartDevice
{
    public void TurnOn() => Console.WriteLine("Light is ON.");
    public void TurnOff() => Console.WriteLine("Light is OFF.");
}

class Fan : SmartDevice
{
    public void TurnOn() => Console.WriteLine("Fan is ON.");
    public void TurnOff() => Console.WriteLine("Fan is OFF.");
}
```

```
class AC : SmartDevice
{
    public void TurnOn() => Console.WriteLine("AC is ON.");
    public void TurnOff() => Console.WriteLine("AC is OFF.");
}

class SmartHomeController
{
    public void Control(SmartDevice device)
    {
        device.TurnOn();
        device.TurnOff();
    }
}
```

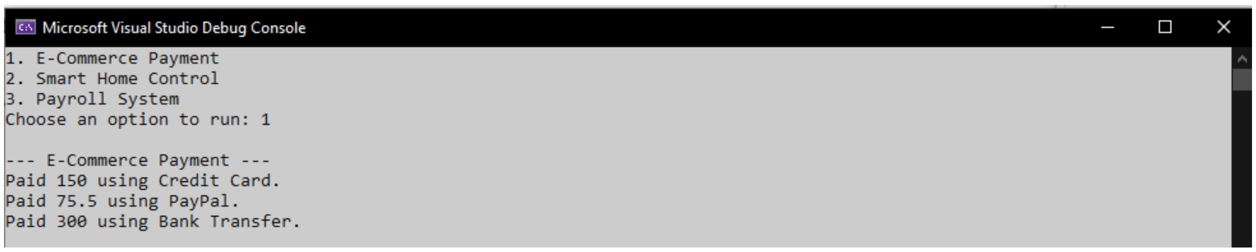
4. Program.cs:

```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        Console.WriteLine("1. E-Commerce Payment");
        Console.WriteLine("2. Smart Home Control");
        Console.WriteLine("3. Payroll System");
        Console.Write("Choose an option to run: ");
        int option = Convert.ToInt32(Console.ReadLine());

        if (option == 1)
        {
           Console.WriteLine("\n--- E-Commerce Payment ---");

           CreditCardPayment creditCard = new CreditCardPayment();
           creditCard.Pay(150);

           PayPalPayment paypal = new PayPalPayment();
           paypal.Pay(75.5);

           BankTransferPayment bank = new BankTransferPayment();
           bank.Pay(300);
        }
        else if (option == 2)
        {
           Console.WriteLine("\n--- Smart Home Control ---");

           Light light = new Light();
           light.TurnOn();
           light.TurnOff();
```

```
    Fan fan = new Fan();
    fan.TurnOn();
    fan.TurnOff();

    AC ac = new AC();
    ac.TurnOn();
    ac.TurnOff();
}
else if (option == 3)
{
    Console.WriteLine("\n--- Payroll System ---");

    FullTimeEmployee fullTime = new FullTimeEmployee(5000);
    Console.WriteLine("Full-Time Employee Pay: " + fullTime.CalculatePay());

    PartTimeEmployee partTime = new PartTimeEmployee(20, 80);
    Console.WriteLine("Part-Time Employee Pay: " + partTime.CalculatePay());

    Contractor contractor = new Contractor(1200);
    Console.WriteLine("Contractor Pay: " + contractor.CalculatePay());
}
else
{
    Console.WriteLine("Invalid option.");
```

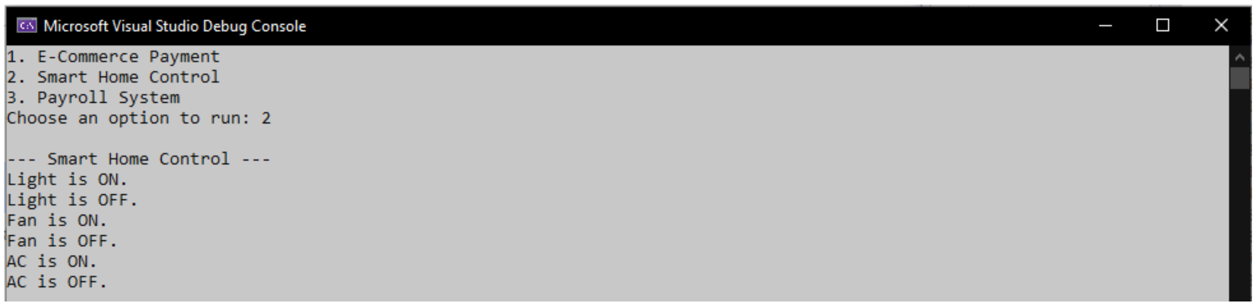
```
        }
    }
}
```

Output:



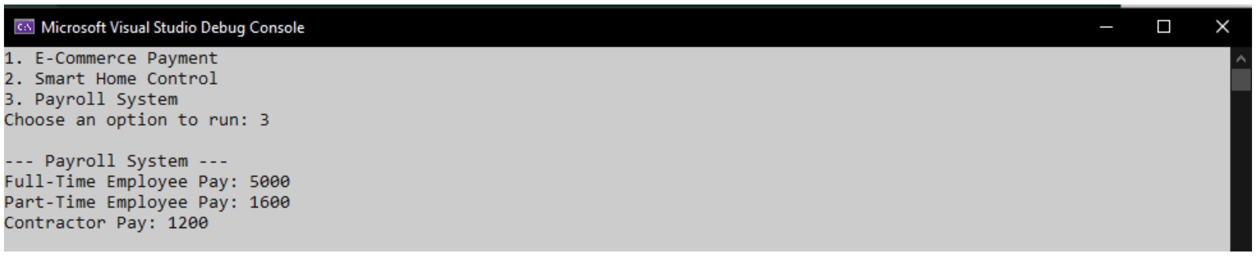
Microsoft Visual Studio Debug Console

```
1. E-Commerce Payment
2. Smart Home Control
3. Payroll System
Choose an option to run: 1

--- E-Commerce Payment ---
Paid 150 using Credit Card.
Paid 75.5 using PayPal.
Paid 300 using Bank Transfer.
```



Microsoft Visual Studio Debug Console

```
1. E-Commerce Payment
2. Smart Home Control
3. Payroll System
Choose an option to run: 2

--- Smart Home Control ---
Light is ON.
Light is OFF.
Fan is ON.
Fan is OFF.
AC is ON.
AC is OFF.
```



Microsoft Visual Studio Debug Console

```
1. E-Commerce Payment
2. Smart Home Control
3. Payroll System
Choose an option to run: 3

--- Payroll System ---
Full-Time Employee Pay: 5000
Part-Time Employee Pay: 1600
Contractor Pay: 1200
```