Name : HEMANT SOPAN PATIL

Class : MSC ICT 2

Practical Assignment – 1

SUBJECT :- .NET CORE

1. **Create a console application using C#.NET Core to demonstrate multi-level inheritance using implementation of access modifiers, properties, methods and constructors. Provide menu to the user to select type of data to store in a Generic List. Use dynamic polymorphism to create Generic List.**

   CODE :-

```
6 references
class Person
{
    2 references
    protected string Name { get; set; }

    1 reference
    public Person(string name)
    {
        Name = name;
    }

    6 references
    public virtual void Display()
    {
        Console.WriteLine("Name: " + Name);
    }
}

4 references
class Student : Person
{
    2 references
    protected int RollNo { get; set; }

    2 references
    public Student(string name, int rollNo) : base(name)
    {
        RollNo = rollNo;
    }

    6 references
    public override void Display()
    {
        base.Display();
        Console.WriteLine("Roll No: " + RollNo);
    }
}
```

```
2 references
class CollegeStudent : Student
{
    2 references
    public string Course { get; set; }

    1 reference
    public CollegeStudent(string name, int rollNo, string course)
        : base(name, rollNo)
    {
        Course = course;
    }

    6 references
    public override void Display()
    {
        base.Display();
        Console.WriteLine("Course: " + Course);
    }
}
```

IMPLEMENTATION IN MAIN CLASS :-

```
List<Person> people = new List<Person>();
people.Add(new CollegeStudent("XYZ", 101, "Cloud Computing"));
people.Add(new Student("ABC", 102));

foreach (var p in people) { p.Display(); Console.WriteLine(); }
break;
```

OUTPUT :-

```
Name: XYZ
Roll No: 101
Course: Cloud Computing

Name: ABC
Roll No: 102
```

2. Create a C#.NET Core console application to demonstrate dynamic polymorphism using Interface type implementation.

CODE :-

```
public interface IPayment
{
    3 references
    void MakePayment();
}

1 reference
public class CreditCardPayment : IPayment
{
    2 references
    public void MakePayment()
    {
        Console.WriteLine("Payment made using Credit Card.");
    }
}

1 reference
public class PaypalPayment : IPayment
{
    2 references
    public void MakePayment()
    {
        Console.WriteLine("Payment made using PayPal.");
    }
}
```

IMPLEMENTATION IN MAIN CLASS :

```
Console.WriteLine("\nSelect Payment Method:");
Console.WriteLine("1. Credit Card");
Console.WriteLine("2. PayPal");
Console.Write("Choice: ");

int payChoice = Convert.ToInt32(Console.ReadLine());

IPayment payment = GetPaymentMethod(payChoice);

payment.MakePayment();
break;
```

OUTPUT :-

```
Select Payment Method:
1. Credit Card
2. PayPal
Choice: 1
Payment made using Credit Card.
```

3. Create a C#.NET Core console application to demonstrate dynamic polymorphism using Generic Interface type implementation.

CODE :-

```
2 references
public class DataManager<T> : IDataStore<T>, IRepository
{
    private List<T> _items = new List<T>();

    3 references
    public void AddItem(T item)
    {
        _items.Add(item);
    }

    2 references
    public void ShowInfo()
    {
        string typeName = typeof(T).Name;
        Console.WriteLine("Currently managing storage for type: " + typeName);
    }

    2 references
    public void DisplayAll()
    {
        Console.WriteLine("--- Listing All Records ---");
        foreach (var item in _items)
        {
            if (item is Person p) p.Display();
            else Console.WriteLine(item);
        }
    }
}
```

IMPLEMENTATION IN MAIN CLASS :-

```
    IDataStore<int> intStore = new DataManager<int>();
    intStore.AddItem(10);
    intStore.AddItem(20);
    intStore.DisplayAll();
    break;
```

OUTPUT :-