Academia International College

(Affiliated to Tribhuvan University)

Lab Report

on

Object Oriented Programming Concept on C#

Submitted By:

Sagar Timalsena

Roll No:15847(23)

Submitted To:

Chandan Bhagat Gupta

Theory:

Object Oriented Programming:

1. OOP is a computer programming model that organizes software design around data, or objects, rather than functions and logic.
2. It is based on the concept of objects and classes.
3. Class is an abstract blueprint used to create more specific, concrete objects. Classes often represent broad categories that share attributes. These classes define what attributes an instance of this type will have, but not the value of those attributes for a specific object.
4. Objects are instances of classes created with specific data.
5. C# is not purely OOP though it is fully OOP because it does not differentiate object and primitive data type.

Fundamental Principle of OOP:

1. Encapsulation: binds together the data and functions that manipulate the data and that keeps both safe from outside interference and misuse.
2. Abstraction: process of hiding the implementation details from the user, only the functionality will be provided to the user.
3. Inheritance: mechanism in which child classes inherit data and behaviors from parent class.
4. Polymorphism: the provision of a single interface to entities of different types of the use of a single symbol to represent multiple different types.

Code:

1. **WAP to show the basics of class and object.**

```
using System;
namespace ClassAndObject
{
    public class Testing
    {
        public void Checking()
        {
           Console.WriteLine("Hello Sagar Timalsena");
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Testing Check123 = new Testing();
            Check123.Checking();
        }
    }
}
```

2. **Create a class and its object to show the inheritance.**

Student.cs

```
using System;
namespace Inheritance
{
    public class Student
    {
        string name;
        int rollNo;

        public void addStudent(string Name, int RollNo)
        {
            name = Name;
            rollNo = RollNo;
        }

        public void personalDetail()
        {
            Console.WriteLine("Hello, this is {0} and my roll no. is {1}", name, 
rollNo);
        }
    }
}
```

```
Program.cs
using System;
namespace Inheritance
{
    class Program
    {
        static void Main(string[] args)
        {
            MarkSheet no1 = new MarkSheet(85,80,70);
            no1.Result();
        }
    }
}
```

3. WAP that reflects the overloading and overriding of constructor and functions.

Overriding of constructor cannot be done because constructors simply are not invoked through derived class and constructors are not inherited from the base class. We can only overload constructor.

Both overloading and overriding of functions can be done. To override the function of base class we must include keyword " virtual" in the function of base class and include keyword "override" in the function of derived class.

For example: public virtual void example(); // Inside Base class public override void example(); // Inside Derived class

Base.cs

```
using System;
namespace OverrideAndOverload
{
    public class Base
    {
        public string Name;

        public Base()
        {
            Console.WriteLine("Welcome Everyone.");
        }

        public Base(string name)
        {
            Name = name;
            Console.WriteLine("Welcome {0}, to base class.",Name);
        }

        public virtual void display()
        {
            Console.WriteLine("This is base class.");
        }
    }
}
```

Derived.cs

```
using System;
namespace OverrideAndOverload
{
    class Program
    {
        static void Main(string[] args)
        {
            // Default Constructor
            Base b1 = new Base();

            // Constructor Overloading
            Base b2 = new Base("Sagar");
             b1.display();
             Derived d1 = new Derived();
            // Function Overriding
            d1.display();
             // Function Overloading
            d1.display("Sagar");
        }
    }
}
```

Conclusion:

Hence, we implemented the basic concept of OOP using C# including the four fundamental principles of OOP: Encapsulation, Abstraction, Inheritance and Polymorphism.

GitHub Repository:

All the above codes used in this report are uploaded in GitHub Repository:

https://github.com/Sagar1555/ncclab