NB Healthcare Technologies Pvt Ltd

Day 10 Morning Assignment (4 – Feb- 2022)

By

Vamsi Krishna Mandapati

1. Write the two points discussed about inheritance in the class.

1. Inheritance is the process of re-using the base class methods in the derived class.
2.Inheritance mail goal is Re-usability. And to remove duplicate code.
Types Of Inheritance:
a. Single Inheritance
b. Multiple Inheritance
c. Multilevel Inheritance.

2. Write example code for:
   a. Single inheritance
   b. Multi level inheritance

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day10Project1
{
    class Algebra
    {
        public int Add(int a, int b)
        {
            return a + b;
        }

        public int Sub(int a, int b)
        {
            return a - b;
        }
    }

    class TotalMaths : Algebra
```

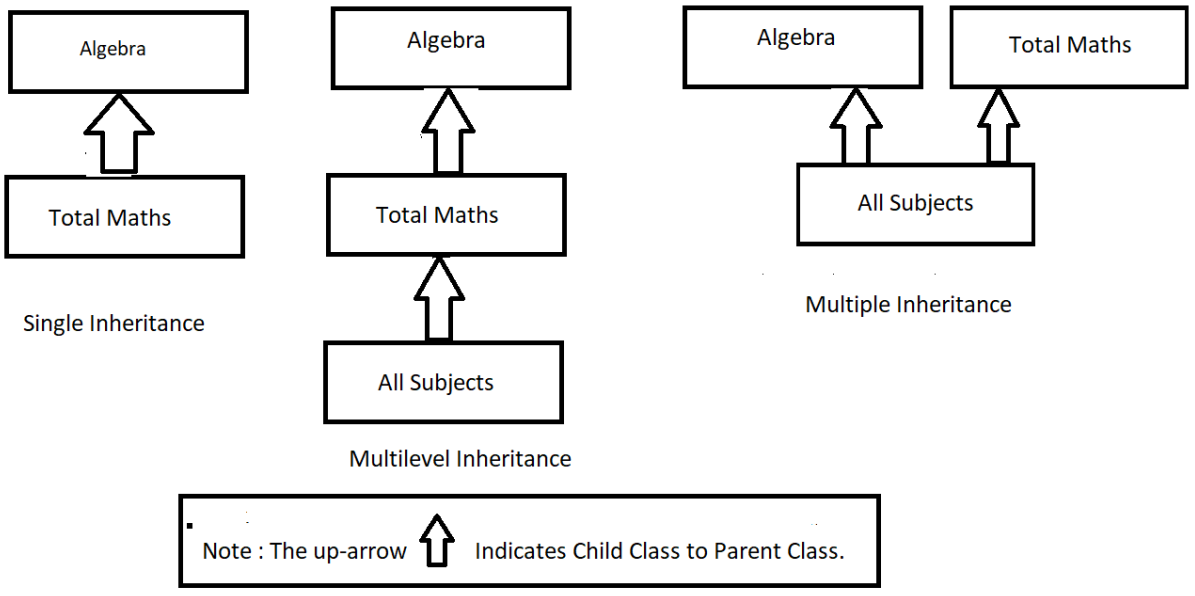
```
    {
        public int Mul(int a, int b)
        {
            return (a * b);
        }
    }

    class AllSubjects : TotalMaths
    {
        public string Water()
        {
            return "H2O";
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            TotalMaths tm = new TotalMaths();
            Console.WriteLine("Single Inheritance");
            Console.WriteLine("Addition: " +
tm.Add(5,4));
            Console.WriteLine("Substraction: " +
tm.Sub(5,4));
            Console.WriteLine("Multiplication: " +
tm.Mul(5,4));


Console.WriteLine("**********************************
*****************************");


            AllSubjects al = new AllSubjects();
            Console.WriteLine("Multilevel Inheritance");
            Console.WriteLine("Addition: " + al.Add(5,
4));
            Console.WriteLine("Substraction: " +
al.Sub(5, 4));
            Console.WriteLine("Multiplication: " +
al.Mul(5, 4));
            Console.WriteLine("Water: " + al.Water());
```

```
            Console.ReadLine();

        }
    }
}
```

Output:

```
System;
D:\NB HealthCare Training\DotNet Projects\Day 9 Morning Assignments\Day10Project1\Day10Project1\bin\Debug\Day10Project1.exe
Single Inheritance
Addition: 9
Substraction: 1
Multiplication: 20
**********************************************************************
Multilevel Inheritance
Addition: 9
Substraction: 1
Multiplication: 20
Water: H2O
```

3. Pictorially represent 3 types of inheritance discussed in the class.



4. Why multiple inheritance is not supported for classes in C#

1. C# does not support **multiple inheritance** , because they reasoned that adding multiple inheritance added too much **complexity** to C# while providing too little benefit. In C#, the classes are only allowed to inherit from a single parent class, which is called **single inheritance** .

2. C# compiler is designed not to support multiple inheritence **because it causes ambiguity of methods from different base class**, If We Have 2base clases and 1 derived class then, derived class method may confuse that from which base class we have to inherit out of two base classes.

5. What is polymorphism.

1. Polymorphism is the ability of an object to take on many forms.
2. If a single entity shows multiple forms or multiple behaviours, then it is called as polymorphism.

Using polymorphism we can achieve flexibility, where a single entity can perform different operations according to the requirement

3.Polymorphism has two types

a. Method Overloading :- Same Method Name With Different Parameters

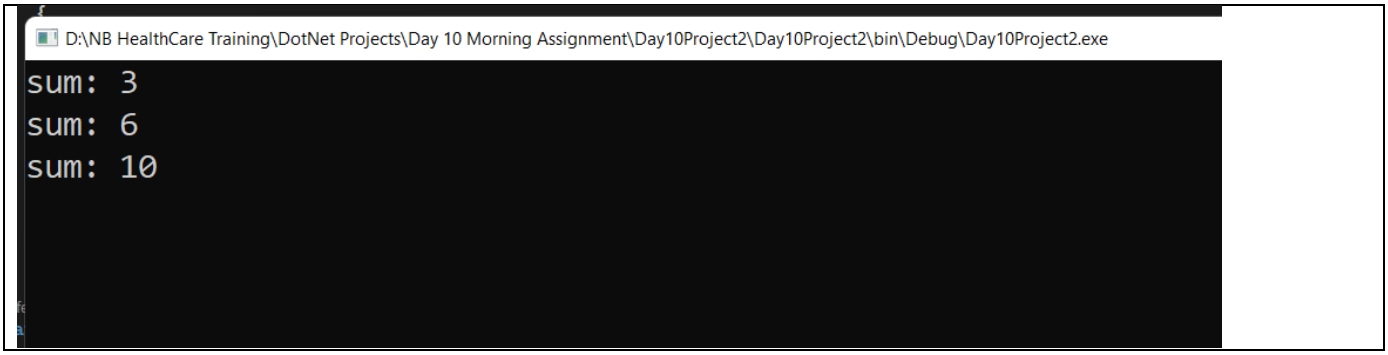
b. Method Overriding :-

6. Write sample code for method overloading.

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day10Project2
{
    internal class Program
    {
        class ArithmeticOperation
        {
            public int Add(int a, int b)
            {
                return a + b;
            }

            public int Add(int a, int b, int c)
            {
                return a + b + c;
            }

            public int Add(int a, int b, int c, int d)
            {
                return a + b + c + d;
            }
        }


        static void Main(string[] args)
        {
            ArithmeticOperation op = new ArithmeticOperation();
            Console.WriteLine("sum: " + op.Add(1,2));
            Console.WriteLine("sum: " + op.Add(1, 2,3));
            Console.WriteLine("sum: " + op.Add(1, 2,3,4));

            Console.ReadLine();
        }
    }
}
```

Output:



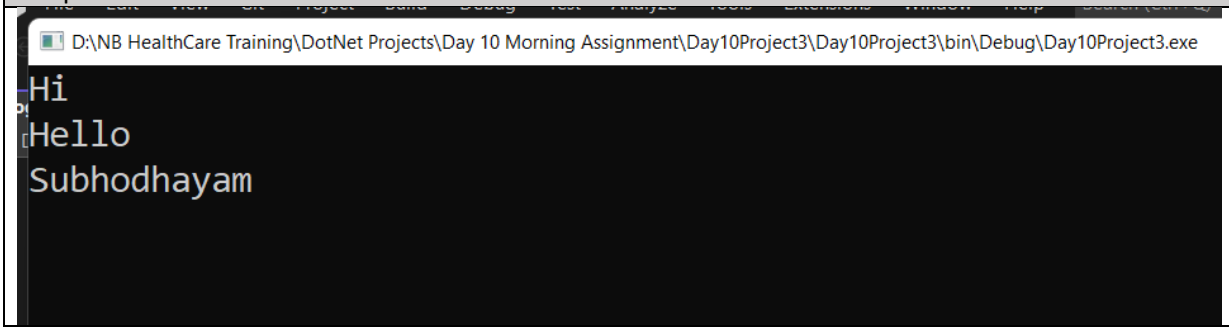
7. Write sample code for method overriding
[using new key word]

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day10Project3
{
    internal class Program
    {
        class EnglishMessage
        {
            public void PrintHi()
            {
                Console.WriteLine("Hi");
            }

            public void PrintHello()
            {
                Console.WriteLine("Hello");
            }

            public void PrintGm()
            {
                Console.WriteLine("Good Morning");
            }
        }

        class TeluguMessage : EnglishMessage
        {
```

```
            public new void  PrintGm()
            {
                Console.WriteLine("Subhodhayam");
            }
        }

        static void Main(string[] args)
        {
            TeluguMessage tl = new TeluguMessage();
            tl.PrintHi();
            tl.PrintHello();
            tl.PrintGm();

            Console.ReadLine();

        }
    }
}
```

Output:



8. Research and write sample code for method overriding using virtual, override keyword.

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day10Project4
{
    internal class Program
    {
        class EnglishMessage
        {
            public void PrintHi()
            {
```

```
                Console.WriteLine("***************Using Virtual-
override***********************");
                Console.WriteLine("Hi");
            }

            public void PrintHello()
            {
                Console.WriteLine("Hello");
            }

            public virtual void PrintGm()
            {
                Console.WriteLine("Good Morning");
            }
        }

        class TeluguMessage : EnglishMessage
        {
            public override void PrintGm()
            {
                Console.WriteLine("Subodhayam");
            }
        }

        static void Main(string[] args)
        {
            TeluguMessage tl = new TeluguMessage();
            tl.PrintHi();
            tl.PrintHello();
            tl.PrintGm();

            Console.ReadLine();
        }
    }
}
```

Output: