

Day 10 Morning Assignment

By

VARUN SAI KUMAR CHEGONI

NB Healthcare and Technology

Date: 04 Feb 2022

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

Answer:

- Inheritance is the process of re-using base class methods in the derived class.
- Inheritance main goal is: Re-usability and to remove duplicate code.

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 ExCodeInheritance
{
    class MathsBasics
    {
        public int a = 6;
        public int b = 2;
        public int AddNumbers()
        {
            return a + b;
        }
        public int DiffNumbers()
        {
            return a - b;
        }
    }

    class MathsComplex : MathsBasics
    {
        public int ProdNumbers()
        {
            return a * b;
        }
        public int DivNumbers()
        {
            return a / b;
        }
    }
}
```

```

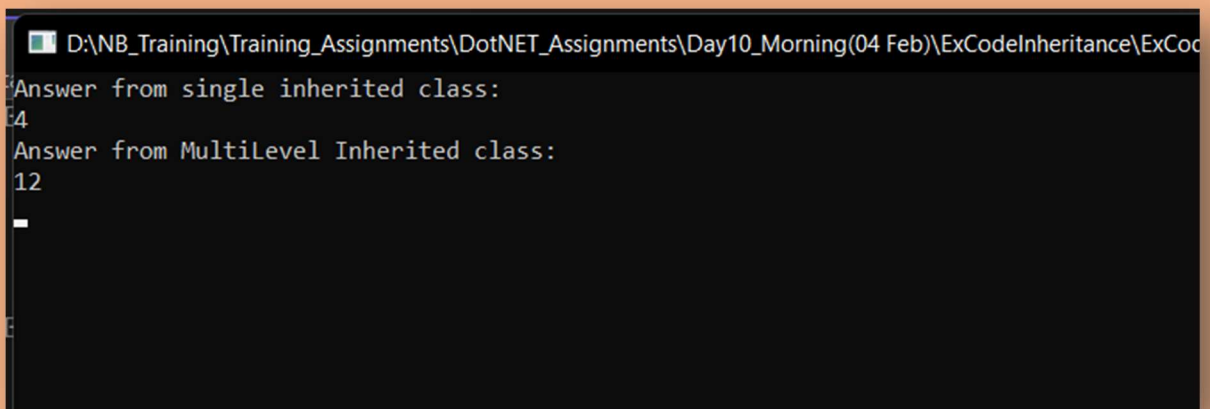
class MathsOperation : MathsComplex
{
    public int Mod()
    {
        return a % b;
    }
}
internal class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Answer from single inherited class: ");
        MathsComplex ans = new MathsComplex();
        Console.WriteLine(ans.DiffNumbers());

        Console.WriteLine("Answer from MultiLevel Inherited class: ");
        MathsOperation ans1 = new MathsOperation();
        Console.WriteLine(ans.ProdNumbers());

        Console.ReadLine();
    }
}

```

Output :



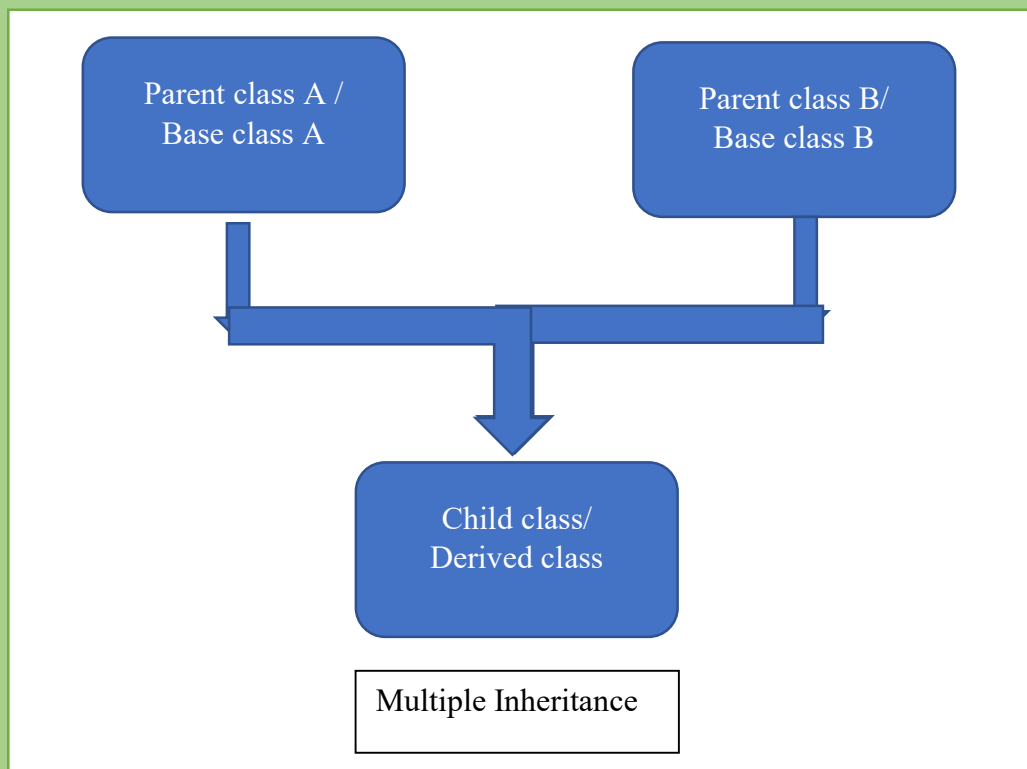
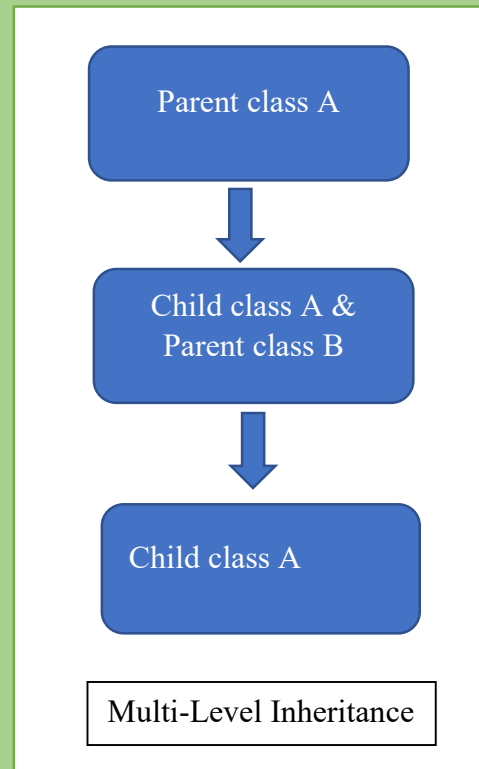
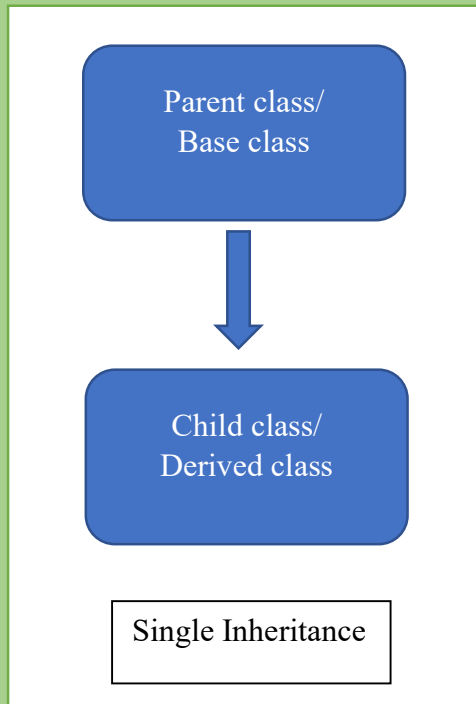
```

D:\NB_Training\Training_Assignments\DotNET_Assignments\Day10_Morning(04 Feb)\ExCodeInheritance\ExCoc
Answer from single inherited class:
4
Answer from MultiLevel Inherited class:
12
_

```

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

Answer:



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

Answer:

1. C# compiler is designed not to support multiple inheritance because it causes ambiguity of methods from different base class.
2. Diamond problem which is associated with multiple class inheritance. For example, I am having two classes namely class2 and class3 and these two classes are inherited from class1. Now we have another class namely class4 which is inherited from both class2 and class3. If the method in class4 calls a method in class1 and class 4 has not overridden the invoked method. Both class2 and class3 has overridden the same method differently. So there occurs the ambiguity problem while invoking the methods. In order to solve this problem C# does not support multiple class inheritance.

5. What is polymorphism.

Answer:

- Polymorphism is a Greek word, meaning "one name many forms".
- In other words, one object has many forms or has one name with multiple functionalities.
- "Poly" means many and "morph" means forms.
- Polymorphism provides the ability to a class to have multiple implementations with the same name.
- It is one of the core principles of Object-Oriented Programming after encapsulation and inheritance.
- There are two types of polymorphism as below:
 - Method Overloading.
 - 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;

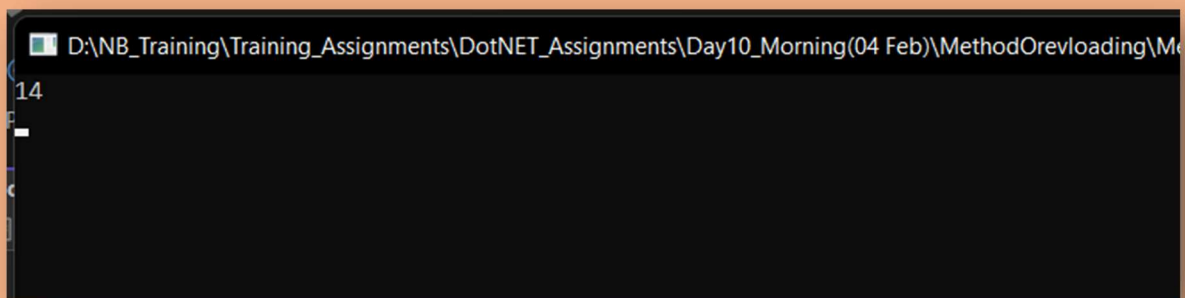
namespace MethodOrevloading
{
    public class Overloading
    {
        public int Add(int a, int b)
        {
            return a + b;
        }

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

    internal class Program
    {
        static void Main(string[] args)
        {
            Overloading obj = new Overloading();
            Console.WriteLine(obj.Add(2,4,8));

            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;
```

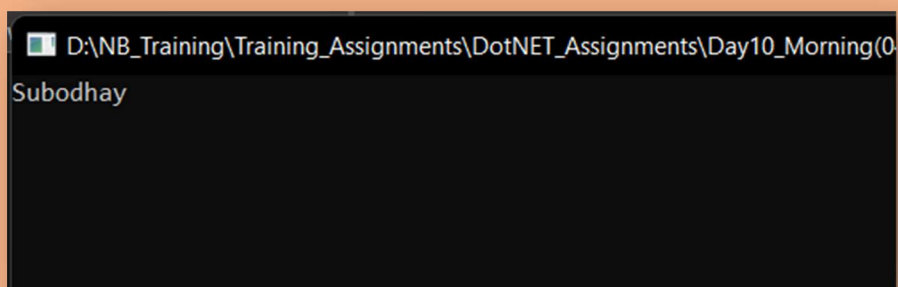
```

namespace OverridingusingNew
{
    class EnglishMsg
    {
        public void PrintHi()
        {
            Console.WriteLine("Hi");
        }
        public void PrintHello()
        {
            Console.WriteLine("Hello");
        }
        public void PrintGM()
        {
            Console.WriteLine("Good Morning");
        }
    }
    class HindiMsg : EnglishMsg
    {
        public new void PrintGM()
        {
            Console.WriteLine("Subodhay");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            HindiMsg msg = new HindiMsg();
            msg.PrintGM();

            Console.ReadLine();
        }
    }
}

```

Output :



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

Code :

Output :

