



DAY10 MORNING ASSIGNMENT

Bhanu Prakash Reddy



FEBRUARY 4, 2022
NB HEALTHCARE TECHNOLOGIES

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

○ **INHERITANCE**

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

Types of Inheritance:

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance

2. Write example code for:

 - a. Single inheritance
 - b. Multi level inheritance

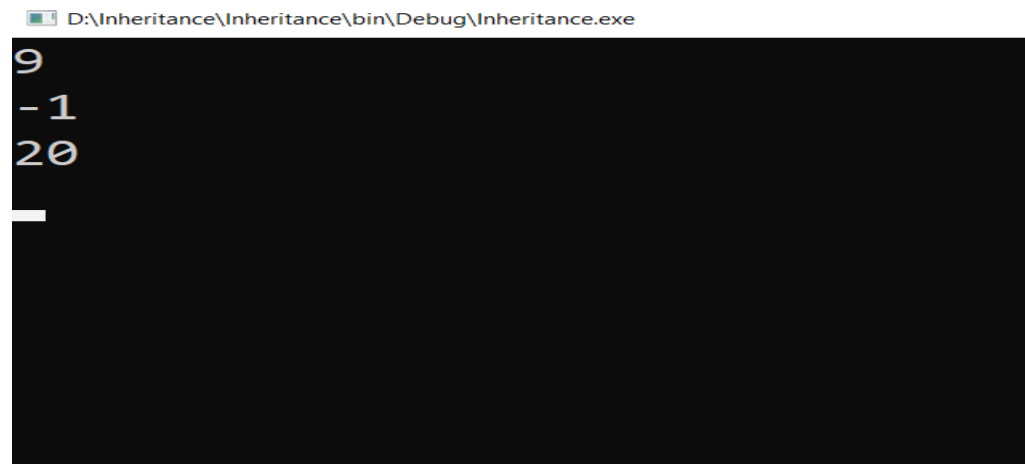
Code: a) single inheritance

```
class Algebra
{
    /// <summary>
    /// Algebra
    /// </summary>
    /// <param name="a"></param>
    /// <param name="b"></param>
    /// <returns>sum,sub</returns>
    public int Add(int a, int b)
    {
        return a + b;
    }
    public int Sub(int a, int b)
    {
        return a - b;
    }
}
class TotalMaths : Algebra
{
    /// <summary>
    /// TotalMaths
    /// </summary>
    /// <param name="a"></param>
    /// <param name="b"></param>
    /// <returns>sum,sub,Multiplication</returns>
    public int Mul(int a, int b)
    {
        return (a * b);
    }
}

internal class Program
{
    static void Main(string[] args)
    {
        TotalMaths tm = new TotalMaths();
        Console.WriteLine(tm.Add(4, 5));
        Console.WriteLine(tm.Sub(4, 5));
        Console.WriteLine(tm.Mul(4, 5));

        Console.ReadLine();
    }
}
```

Output:



```
D:\Inheritance\Inheritance\bin\Debug\Inheritance.exe
9
- 1
20

```

b) multilevel Inheritance

```
class Algebra
{
    /// <summary>
    /// Algebra
    /// </summary>
    /// <param name="a"></param>
    /// <param name="b"></param>
    /// <returns>sum,sub</returns>
    public int Add(int a, int b)
    {
        return a + b;
    }
    public int Sub(int a, int b)
    {
        return a - b;
    }
}
class TotalMaths : Algebra
{
    /// <summary>
    /// TotalMaths
    /// </summary>
    /// <param name="a"></param>
    /// <param name="b"></param>
    /// <returns>sum,sub,Multiplication</returns>
    public int Mul(int a, int b)
    {
        return (a * b);
    }
}
class AdvMaths : TotalMaths
{
    /// <summary>
    /// AdvMaths
    /// </summary>
    /// <param name="a"></param>

```

```

    /// <param name="b"></param>
    /// <returns>sum,sub,Multiplication,Division</returns>
    public int Div(int a,int b)
    {
        return a / b;
    }
}


internal class Program
{
    static void Main(string[] args)
    {
        AdvMaths am = new AdvMaths();
        Console.WriteLine(am.Add(32, 6));
        Console.WriteLine(am.Sub(32, 6));
        Console.WriteLine(am.Mul(32, 6));
        Console.WriteLine(am.Div(32, 6));

        Console.ReadLine();

    }
}

```

Output:

 D:\MultiLevel Inheritance\MultiLevel Inheritance\bin\Debug\MultiLevel Int

```

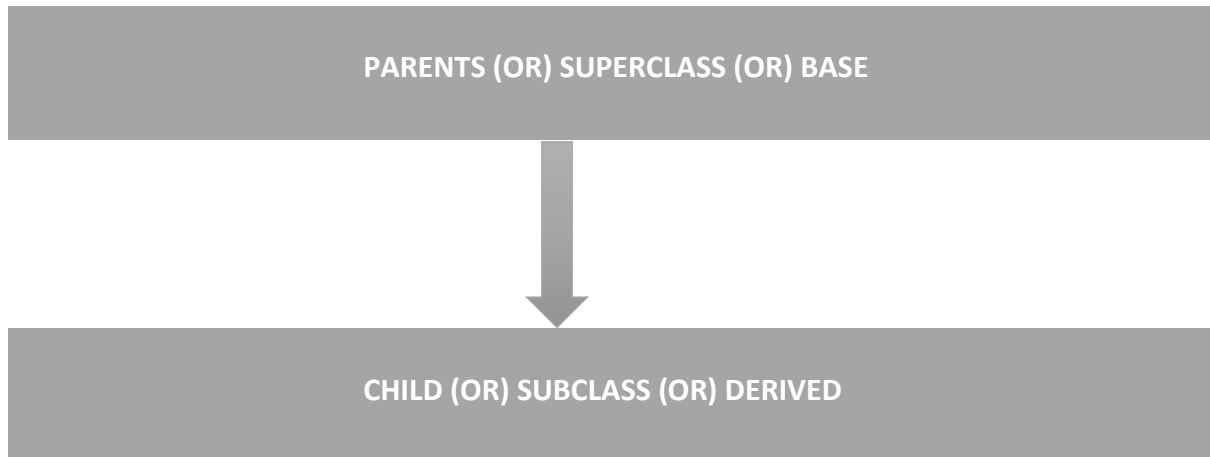
38
26
192
5

```

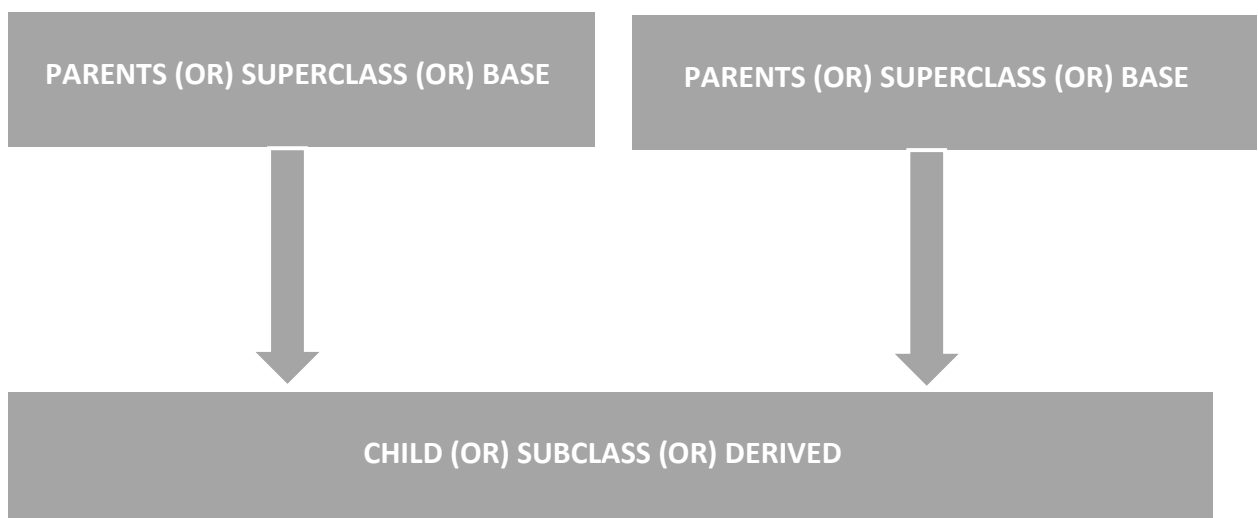
3. Pictorially represent 3 types of inheritance discussed in the class.

PICTORIAL REPRESENTATION OF TYPES OF INHERITENCE

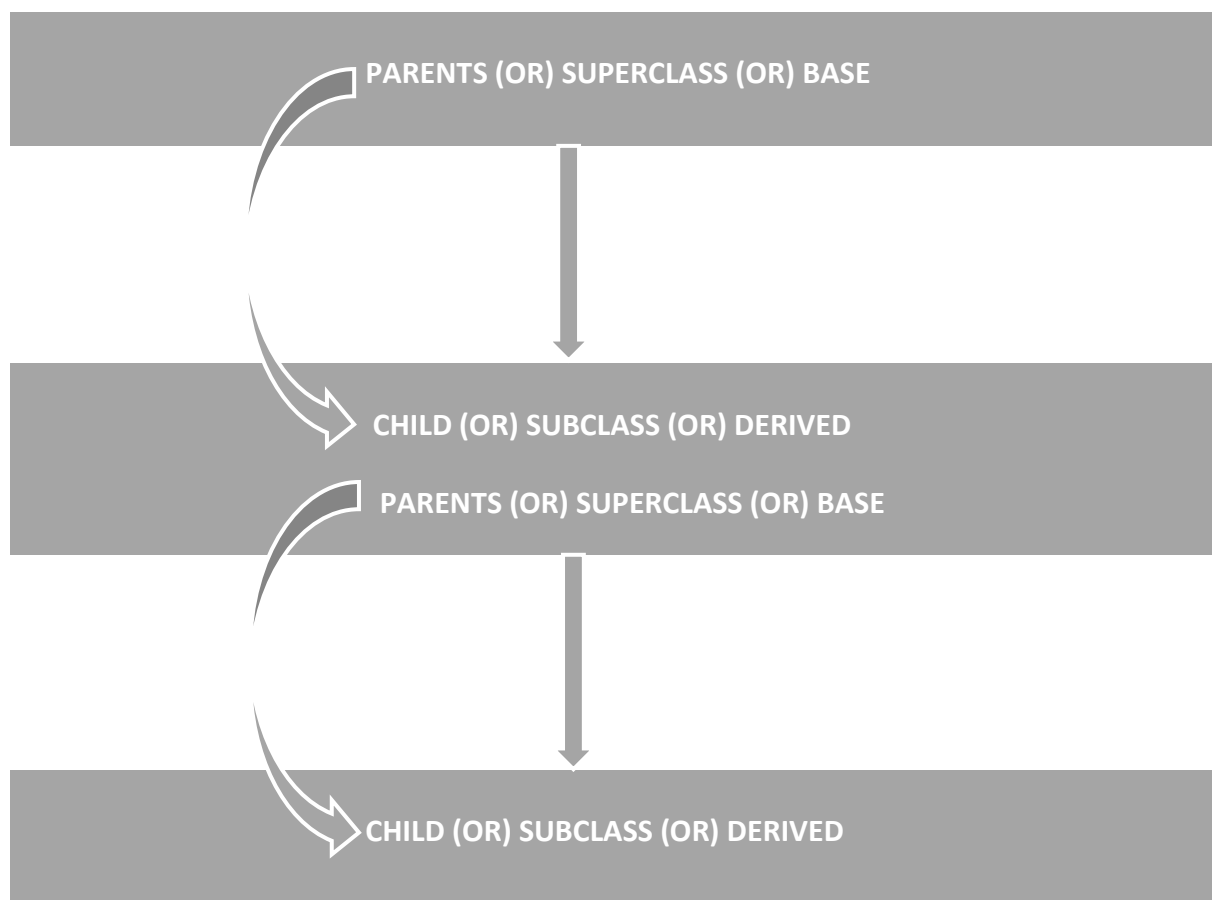
Single Inheritance:



MULTIPLE INHERITANCE:



MULTI-LEVEL INHEITENCE:



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

- In Multiple inheritance, one class can have more than one superclass and inherit features from all its parent classes. ... But C# does not support multiple class inheritance.
- One of the main reasons behind this is the “diamond problem”.
- 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 methods differently. So there occurs the ambiguity problem while invoking the methods.

5. What is polymorphism.

- Polymorphism is the ability of an object to take on many forms.
- Types:

Method Overloading

Method Overriding

- **Method Overloading:** Within the same class, writing same method name with different parameters is called Method Overloading.
- **Method Overriding:** Override a base class method in the derived class by creating a method with the same name and signatures to perform a different task is called Method Overriding.

6. Write sample code for method overloading.

Code:

```
class Algebra
{
    /// <summary>
    /// adding two numbers
    /// </summary>
    /// <param name="a"></param>
    /// <param name="b"></param>
    /// <returns>a+b</returns>
    public int Add(int a, int b)
    {
        return a + b;
    }
    /// <summary>
    /// adding three numbers
    /// </summary>
    /// <param name="a"></param>
    /// <param name="b"></param>
    /// <param name="c"></param>
    /// <returns>a+b+c</returns>
    public int Add(int a, int b, int c)
    {
```




```

        return a + b + c;
    }
    /// <summary>
    /// adding four numbers
    /// </summary>
    /// <param name="a"></param>
    /// <param name="b"></param>
    /// <param name="c"></param>
    /// <param name="d"></param>
    /// <returns>a+b+c+d</returns>
    public int Add(int a, int b, int c, int d)
    {
        return a + b + c + d;
    }
}
static void Main(string[] args)
{
    Algebra al = new Algebra();
    Console.WriteLine(al.Add(34, 46));
    Console.WriteLine(al.Add(24, 76, 43));
    Console.WriteLine(al.Add(12, 23, 64, 77));

    Console.ReadLine();
}
}

```

Output

 D:\Method Overloading\Method Overloading\bin\Debug\Method Overloading.exe

```

80
143
176

```

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


Code:

```
class EnglishMessage
{
    /// <summary>
    /// Print Hi
    /// </summary>
    public void PrintHi()
    {
        Console.WriteLine("Hi");
    }
    /// <summary>
    /// Print Hello
    /// </summary>
    public void PrintHello()
    {
        Console.WriteLine("Hello");
    }
    /// <summary>
    /// Print Good Morning
    /// </summary>
    public void PrintGM()
    {
        Console.WriteLine("Good Morning");
    }
}
class TeluguMessage : EnglishMessage
{
    /// <summary>
    /// Print Subhodhayam
    /// </summary>
    public new void PrintGM()
    {
        Console.WriteLine("Subhodhaym");
    }
}
internal class Program
{
    static void Main(string[] args)
    {
        EnglishMessage em = new EnglishMessage();
        em.PrintHi();
        em.PrintHello();
        em.PrintGM();

        TeluguMessage tm = new TeluguMessage();
        tm.PrintHi();
        tm.PrintHello();
        tm.PrintGM();

        Console.ReadLine();
    }
}
```

Output :

 D:\Method Overriding\Method Overriding\bin\Debug\Method Overriding.exe

Hi

Hello

Good Morning

Hi

Hello

Subhodhaym

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

Code:


```
internal class Program
{
    class EnglishMessage
    {
        /// <summary>
        /// Print Hi
        /// </summary>
        public void PrintHi()
        {
            Console.WriteLine("Hi");
        }
        /// <summary>
        /// Print Hello
        /// </summary>
        public void PrintHello()
        {
            Console.WriteLine("Hello");
        }
        /// <summary>
        /// Print Good Morning
        /// </summary>
        public virtual void PrintGM()
        {
            Console.WriteLine("Good Morning");
        }
    }
    class HindiMessage : EnglishMessage
    {
        /// <summary>
        /// Print Subh Prabhaat
        /// </summary>
        public override void PrintGM()
        {
            Console.WriteLine("Subh Prabhaat");
        }
    }

    static void Main(string[] args)
    {
        EnglishMessage em = new EnglishMessage();
        em.PrintHi();
        em.PrintHello();
        em.PrintGM();

        HindiMessage hm = new HindiMessage();
        hm.PrintHi();
        hm.PrintHello();
        hm.PrintGM();

        Console.ReadLine();
    }
}
```

Output:

 D:\Virtual and Override Keyword\Virtual and Override Keyword\bin\Debug\Virtual and Override Keyword.exe

Hi

Hello

Good Morning

Hi

Hello

Subh Prabhaat