Day10 Morning Assignment

By

Narala Praveen

04-02-2022

Question1: Write the two points discussed about inheritance in the class
Point1: Inheritance is the process of reusing base class methods in derived class.
Point2: Main goal of inheritance is Re-usability and to remove duplicate code.

Question2: Write Example code for :

a. Single inheritance

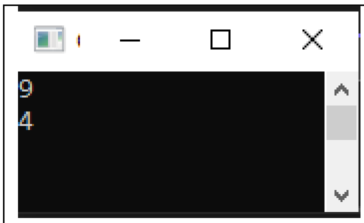
b. Multilevel inheritance.

Single inheritance:

Code:

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

namespace Day10singleinheritance
{
    //******************************************************************\\
    //Author:Narala Praveen
    //Purpose:Simple Code for Single inheritance
    //*******************************************************************\\

    class Algebra
    {
        /// <summary>
        /// This method is for Addition
        /// </summary>
        /// <returns></returns>
        public int Add(int a ,int b)
        {
            return a + b;
        }

    }
    class Substract:Algebra
    {
        /// <summary>
        /// This method SubStract
        /// </summary>
        /// <returns></returns>
        public int Sub(int a ,int b)
        {
            return a - b;
        }

    }


    internal class Program
    {
        static void Main(string[] args)
        {
            Substract obj= new Substract();
            Console.WriteLine(obj.Add(5,4));
            Console.WriteLine(obj.Sub(6,2));

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

Output:

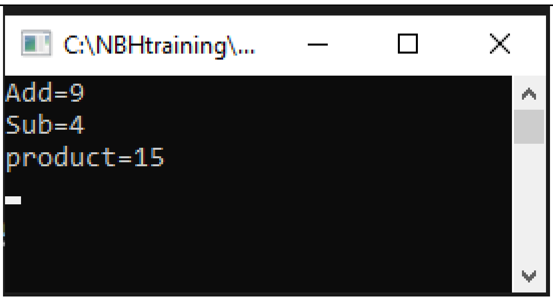


```
9
4
```

MultiLevel Inheritance:

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

namespace Day10Multilevelinheritance
{
    //*******************************************************************\\
    //Author:Narala Praveen
    //Purpose:Simple Code for Multilevel inheritance
    //********************************************************************\\

    class Algebra
    {
        /// <summary>
        /// This method is for Addition
        /// </summary>
        /// <returns></returns>
        public int Add(int a, int b)
        {
            return a + b;
        }

    }
    class Substract : Algebra
    {
        /// <summary>
        /// This method SubStract
        /// </summary>
        /// <returns></returns>
        public int Sub(int a, int b)
        {
            return a - b;
        }

    }
    class Product:Substract
    {
        /// <summary>
        /// This Method is for Multiplication
        /// </summary>
        /// <returns></returns>
        public int Multiplication(int a,int b)
        {
            return a * b;
        }

    }


    internal class Program
    {
        static void Main(string[] args)
        {
            Product obj = new Product();
            Console.WriteLine($"Add={obj.Add(5, 4)}");
            Console.WriteLine($"Sub={obj.Sub(6, 2)}");
            Console.WriteLine($"product={obj.Multiplication(5,3)}");

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

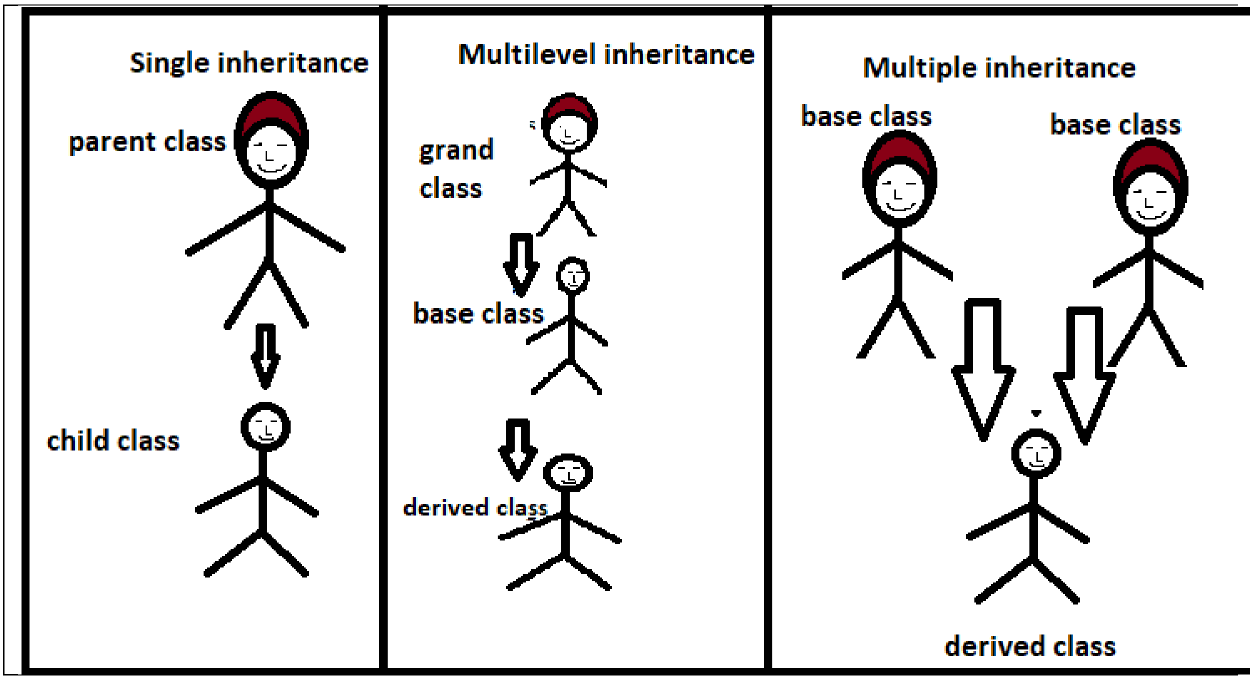
Output:



```
Add=9
Sub=4
product=15
```

Question3:

Pictorially represent 3 types of inheritance discussed in the class?



Question4: Why Multiple Inheritance is not supported for classes in C#?

Reasons:
1.Multiple inheritance causes ambiguity(the quality of being open to more than one interpretation) of methods from different base class.
2. This is the due to the diamond shape of two classes, if two classes Praveen class and Naveen class inherit from Raja class and Rani class inherits from both Praveen and Naveen and so on ……Multiple inheritance is not possible in c#.
3. The above method leads to deadly diamond problem.

Question5: What is Polymorphism?

Definition: polymorphism is defined as the ability of an object to take on many forms is called as Polymorphism.

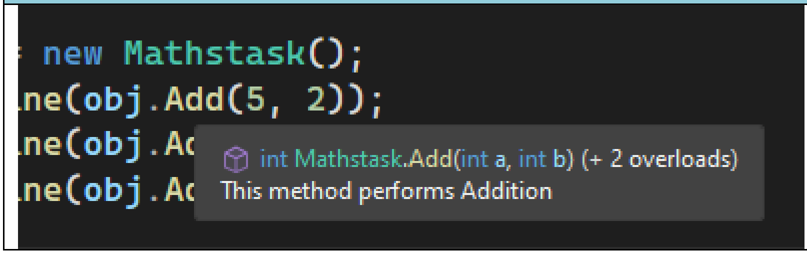
Types:

a. **Method overload.**
b. **Method override.**

Question6:
Write a simple code for Method Overloading?

Code:

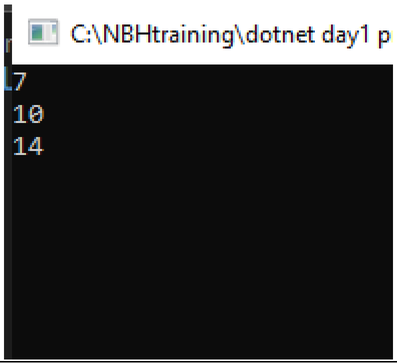
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day10overloadproject
{         //****************************************************************\\
          //Author:Narala Praveen
          //Purpose:To create class and perform method overload
          //*****************************************************************\\

    class Mathstask
    {
        /// <summary>
        /// This method performs Addition
        /// </summary>
        /// <returns></returns>
        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;
        }

    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Mathstask obj = new Mathstask();
            Console.WriteLine(obj.Add(5, 2));
            Console.WriteLine(obj.Add(5,2,3));
            Console.WriteLine(obj.Add(5,2,3,4));

            Console.ReadLine();

        }
    }
}
```



```
new Mathstask();
ne(obj.Add(5, 2));
ne(obj.Ac
ne(obj.Ac
```

int Mathstask.Add(int a, int b) (+ 2 overloads)
This method performs Addition

Output:



C:\NBHtraining\dotnet day1 p

```
7
10
14
```

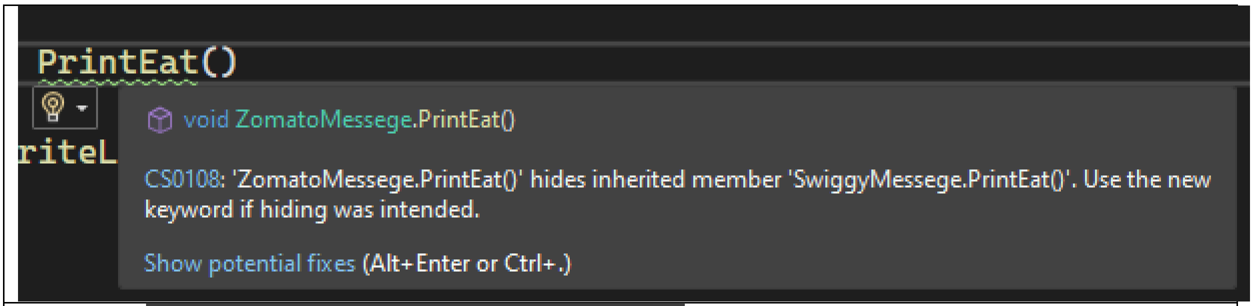
Question7;

Write a simple code for Method overriding(using new Keyword)?

Code:

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

namespace Day10Methodoverriding
{
          //****************************************************************\\
          //Author:Narala Praveen
          //Purpose:To do method overriding using new key word
          //*******************************************************************\\

    class SwiggyMessege
    {/// <summary>
    /// This Method is for Printing
    /// </summary>
        public void printHi()
        {
            Console.WriteLine("Hi");
        }
        public void PrintHello()
        {
            Console.WriteLine("Hello");
        }
        public void PrintEat()
        {
            Console.WriteLine("Eat");
        }
    }

    class ZomatoMessege:SwiggyMessege
    {

        public new void  PrintEat()
        {
            Console.WriteLine("Tinavara");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            ZomatoMessege obj = new ZomatoMessege();

        }
    }
}
```

Suggestion:

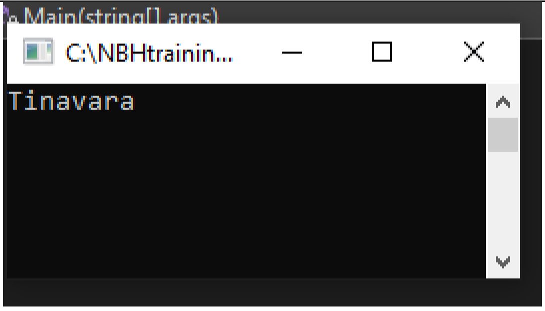


```
PrintEat()
riteL
```

void ZomatoMessege.PrintEat()

CS0108: 'ZomatoMessege.PrintEat()' hides inherited member 'SwiggyMessege.PrintEat()'. Use the new keyword if hiding was intended.

Show potential fixes (Alt+Enter or Ctrl+.)



```
Tinavara
```

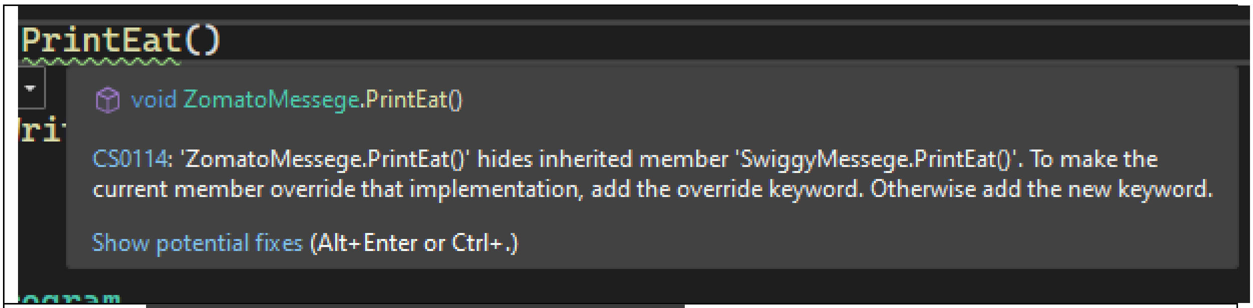
Output:

Question8:

Write a simple code for Method overriding (using virtue, override key word)

Code:

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

namespace Day10Methodoverriding
{
    //****************************************************************\\
    //Author:Narala Praveen
    //Purpose:To do method overriding using Virtue and override keyword
    //*******************************************************************\\

    class SwiggyMessege
    {/// <summary>
     /// This Method is for Printing
     /// </summary>
        public void printHi()
        {
            Console.WriteLine("Hi");
        }
        public void PrintHello()
        {
            Console.WriteLine("Hello");
        }
        public virtual void PrintEat()
        {
            Console.WriteLine("Eat");
        }
    }

    class ZomatoMessege : SwiggyMessege
    {

        public override void PrintEat()
        {
            Console.WriteLine("Tinavara");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            ZomatoMessege obj = new ZomatoMessege();
            obj.PrintEat();
            Console.ReadLine();

        }
    }
}
```

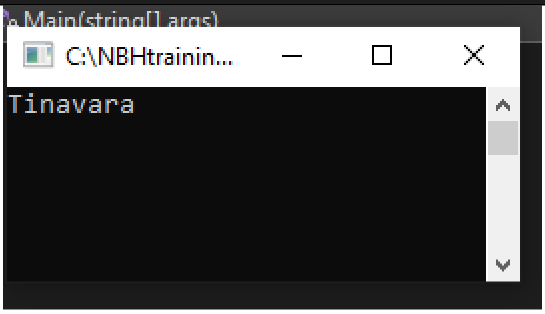


```
PrintEat()
```

void ZomatoMessege.PrintEat()

CS0114: 'ZomatoMessege.PrintEat()' hides inherited member 'SwiggyMessege.PrintEat()'. To make the current member override that implementation, add the override keyword. Otherwise add the new keyword.

Show potential fixes (Alt+Enter or Ctrl+.)



```
Tinavara
```

Output: