Day 10(04-02-2022) Morning Assignment
By
Sudha Kumari Sugasani

Q1.Write the two points discussed about the Inheritance in class.

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

Q2.Write example code for
a.Single Inheritance
b.Multilevel Inheritance

Code:

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

namespace Day10project1
{
    /*****************************************************
     * Author:Sudha Sugasani
     * Purpose:Example code for Single Inheritance
     *****************************************************/
    class Algebra
    {
        /// <summary>
        /// This method do addition of two numbers
        /// </summary>
        /// <param name="a">int</param>
        /// <param name="b">int</param>
        /// <returns>sum</returns>
        public int Add(int a,int b)
        {
            return a + b;
        }
        /// <summary>
        /// This method do Substraction of two numbers
        /// </summary>
        /// <param name="a">int</param>
        /// <param name="b">int</param>
        /// <returns>difference</returns>
        public int Sub(int a,int b)
        {
            return a - b;
        }
    }
    class Totalmaths:Algebra
    {
        /// <summary>
        /// This method do Multiplication of two numbers
        /// </summary>
        /// <param name="a">int</param>
        /// <param name="b">int</param>
        /// <returns>Product</returns>
        public int Mul(int a,int b)
        {
            return (a * b);
```

```
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Totalmaths t1 = new Totalmaths();
            Console.WriteLine($"The Sum of two numbers is {t1.Add(10,15)}");
            Console.WriteLine($"The difference between two numbers is
{t1.Sub(15, 10)}");
            Console.WriteLine($"The product of two numbers is {t1.Mul(10,
5)}");
            Console.ReadLine();

        }
    }
}
```

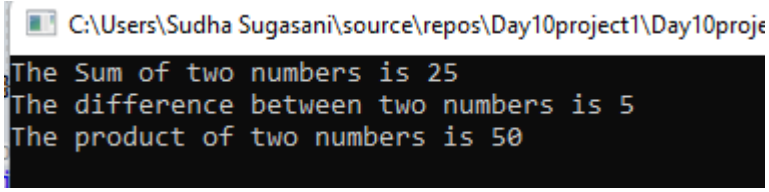
b.

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

namespace Day10project2
{
    using System;
    using System.Collections.Generic;
    using System.Linq;
    using System.Text;
    using System.Threading.Tasks;


            /*****************************************************
             * Author:Sudha Sugasani
             * Purpose:Example code for Multilevel Inheritance,
             *****************************************************/
            class Algebra
            {
                /// <summary>
                /// This method do addition of two numbers
                /// </summary>
                /// <param name="a">int</param>
                /// <param name="b">int</param>
                /// <returns>sum</returns>
                public int Add(int a, int b)
                {
                    return a + b;
                }
                /// <summary>
                /// This method do Substraction of two numbers
                /// </summary>
                /// <param name="a">int</param>
                /// <param name="b">int</param>
                /// <returns>difference</returns>
                public int Sub(int a, int b)
                {
                    return a - b;
                }
```

```
            }
            class Totalmaths : Algebra
            {
                /// <summary>
                /// This method do Multiplication of two numbers
                /// </summary>
                /// <param name="a">int</param>
                /// <param name="b">int</param>
                /// <returns>Product</returns>
                public int Mul(int a, int b)
                {
                    return (a * b);
                }
            }
           class Allsubjects:Totalmaths
            {
                /// <summary>
                /// This method will return formula for Methane
                /// </summary>
                /// <returns>CH4</returns>
               public string Water()
                {
                    return "H2O";
                 }
             }

            internal class Program
            {
                static void Main(string[] args)
                {
                    Allsubjects a1 = new Allsubjects();
                    Console.WriteLine($"The Sum of two numbers is {a1.Add(10,
15)}");
                    Console.WriteLine($"The difference between two numbers is
{a1.Sub(15, 10)}");
                    Console.WriteLine($"The product of two numbers is
{a1.Mul(10, 5)}");
                    Console.WriteLine($"Formula for Methane is {a1.Water()}");
                    Console.ReadLine();

                }
            }


        }
```

Output:

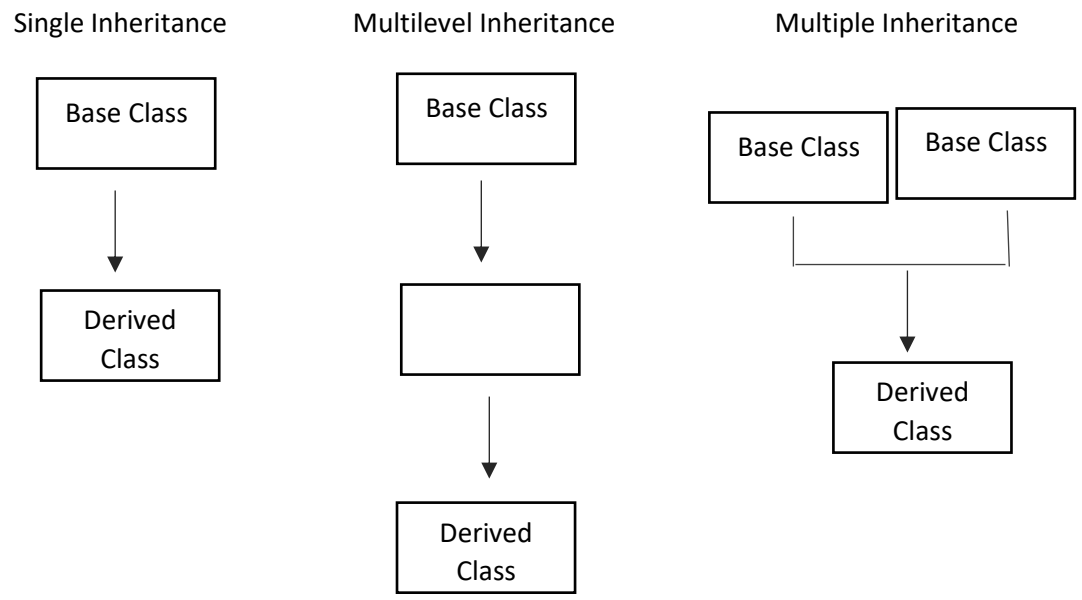
a.



b.

```
C:\NH\.NET Projects\Day10project2\Day10project2\bin\Debug\
The Sum of two numbers is 25
The difference between two numbers is 5
The product of two numbers is 50
Formula for Methane is H2O
```

Q3.Pictorially represent three types of Inheritance discussed in the class.



Q4.Why Multiple Inheritance is not possible in C# through classes

C# compiler is designed not to support Multiple Inheritance because it causes ambiguity of methods from different base classes.
EX. 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 clas3 has overriden the same methos differently. So there occurs the ambiguity problem wile invoking the methods.

Q5.What is Polymorphism.

Polymorphism is the ability of an object to take on many forms.
In Polymorphism there are two types
a.Method Overloading
b.Method Overriding

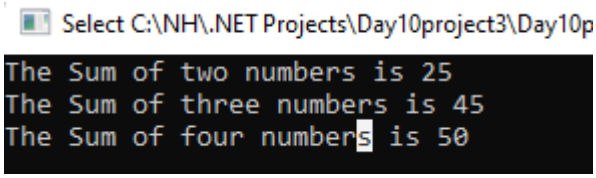
Q6.Write sample code for Method Overloading

Code:

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

```
namespace Day10project3
{
        /*****************************************************
         * Author:Sudha Sugasani
         * Purpose:Example code for Method Overloading
         *****************************************************/

        class Algebra
        {
            /// <summary>
            /// This method do addition of two numbers
            /// </summary>
            /// <param name="a">int</param>
            /// <param name="b">int</param>
            /// <returns>sum</returns>
            public int Add(int a, int b)
            {
                return a + b;
            }
            /// <summary>
            /// This method do addition  of three numbers
            /// </summary>
            /// <param name="a">int</param>
            /// <param name="b">int</param>
            /// <returns>sum</returns>
            public int Add(int a, int b, int c)
            {
                return a + b + c;
            }
            /// <summary>
            /// This method do addition of four numbers
            /// </summary>
            /// <param name="a">int</param>
            /// <param name="b">int</param>
            /// <param name="c">int</param>
            /// <param name="d">int</param>
            /// <returns>sum</returns>
            public int Add(int a, int b, int c, int d)
            {
                return a + b + c + d;
            }
        }


        internal class Program
        {
            static void Main(string[] args)
            {
                Algebra a1 = new Algebra();
                Console.WriteLine($"The Sum of two numbers is {a1.Add(10,
15)}");
                Console.WriteLine($"The Sum of three numbers is {a1.Add(15, 10,
20)}");
                Console.WriteLine($"The Sum of four numbers is {a1.Add(10, 15,
5, 20)}");
                Console.ReadLine();

            }
        }


    }
```

Output:



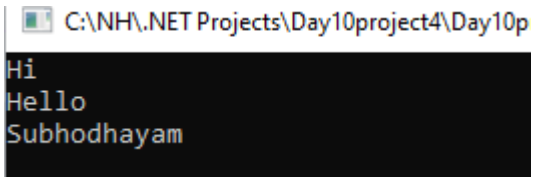
Q7.Write sample code for Method Overriding[using new keyword]

Code:

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

namespace Day10project4
{

        /*****************************************************
         * Author:Sudha Sugasani
         * Purpose:Example code for Method Overriding
         *****************************************************/

        class EnglishMessage
        {
            /// <summary>
            /// This method will print Hi message
            /// </summary>

            public void PrintHi()
            {
            Console.WriteLine("Hi");
            }

            /// <summary>
            /// This method will print Hello Message
            /// </summary>


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


            /// <summary>
            /// This method will print Good Morning message
            /// </summary>
            public void PrintGM()
            {
            Console.WriteLine("Good Morning");
            }
        }
        class TeluguMessage:EnglishMessage
        {
            /// <summary>
            /// This method will print Subhodhayam message
```

```
            /// </summary>
            public new void  PrintGM()
            {
            Console.WriteLine("Subhodhayam");
            }
        }

        internal class Program
        {
            static void Main(string[] args)
            {
                TeluguMessage m1 = new TeluguMessage();
                m1.PrintHi();
                m1.PrintHello();
                m1.PrintGM();

                Console.ReadLine();

            }
        }


    }
```

Output:



Q8.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 Day10project5
{


/*****************************************************************************
         * Author:Sudha Sugasani
         * Purpose:Example code for Method Overriding using virtual,override
keyword

*****************************************************************************/

        class EnglishMessage
        {
            /// <summary>
            /// This method will print Hi message
            /// </summary>

            public void PrintHi()
            {
                Console.WriteLine("Hi");
```

```
        }

        /// <summary>
        /// This method will print Hello Message
        /// </summary>


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


        /// <summary>
        /// This method will print Good Morning message
        /// </summary>
        public virtual void PrintGM()
        {
            Console.WriteLine("Good Morning");
        }
    }
    class TeluguMessage : EnglishMessage
    {
        /// <summary>
        /// This method will print Subhodhayam message
        /// </summary>
        public override void PrintGM()
        {
            Console.WriteLine("Subhodhayam");
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            TeluguMessage m1 = new TeluguMessage();
            m1.PrintHi();
            m1.PrintHello();
            m1.PrintGM();

            Console.ReadLine();

        }
    }


}
```

Output: