Day 10 Morning Assignment

By Manoj Karnatapu - NBHealthCareTechnologies

Assignment 1

Write the two points discussed about inheritance in the class.

Answer

- Inheritance is the process of re-using the base class methods in to the derived class.
- Inheritance main goal is: **Reusability.**
- We Use Inheritance to reduce / eradicate the Code Duplication. Because in Software industry, we prefer DRY Code i.e., DRY - Do Not Repeat.

.

Assignment 2

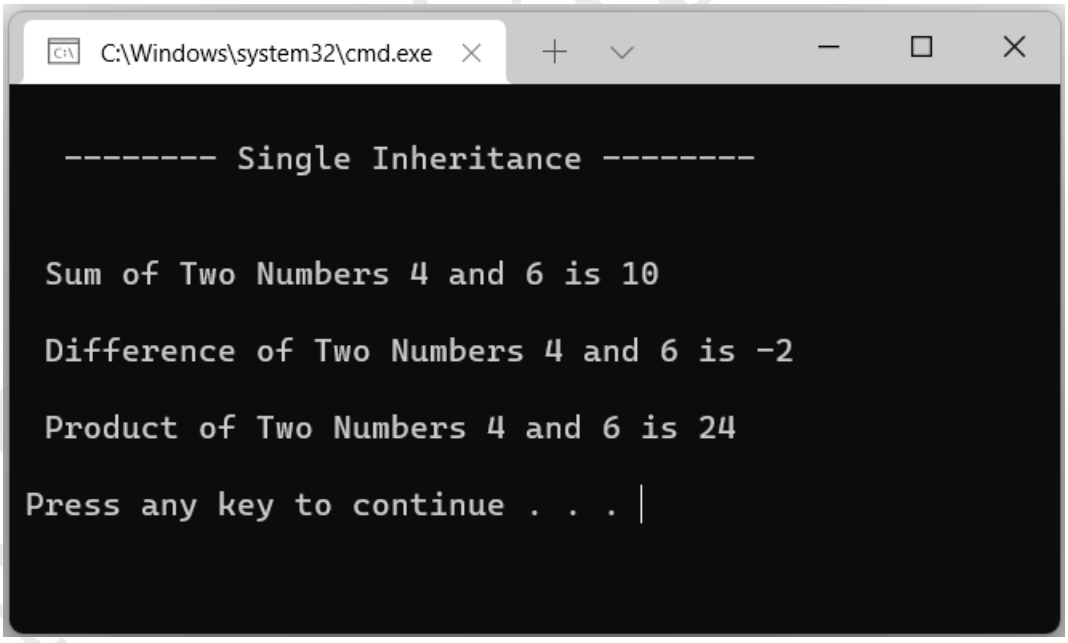
(a). Write Example Code for Single Inheritance.

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Write a C# Code for Single Inheritance

// For Reference, check Day10 Project1 in the same Repository.

namespace Day10Project1
{
    class Algebra
    {
        /// <summary>
        /// This is a Addition Method of Two Numbers
        /// </summary>
        /// <param name="a">int A</param>
        /// <param name="b">int B</param>
        /// <returns>Sum</returns>
        public int Add(int a, int b)
        {
            return a + b;
        }
        /// <summary>
        /// This is a Subtraction Method of Two Numbers
        /// </summary>
        /// <param name="a">int A</param>
        /// <param name="b">int B</param>
        /// <returns>Difference</returns>
        public int Sub(int a, int b)
        {
            return a - b;
        }
    }
    /// <summary>
```

```
    /// Inheriting the Methods of Parent Class Algebra to Child Class TotalMaths.
    /// </summary>
    class TotalMaths : Algebra
    {
        /// <summary>
        /// This is a Multiplication Method of Two Numbers
        /// </summary>
        /// <param name="a">int A</param>
        /// <param name="b">int B</param>
        /// <returns>Product</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("\n  -------- Single Inheritance --------\n");
            Console.WriteLine("\n Sum of Two Numbers 4 and 6 is {0}",tm.Add(4,6));
            Console.WriteLine("\n Difference of Two Numbers 4 and 6 is {0}", tm.Sub(4, 6));
            Console.WriteLine("\n Product of Two Numbers 4 and 6 is {0}", tm.Mul(4, 6));

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

Output



.

Assignment 2

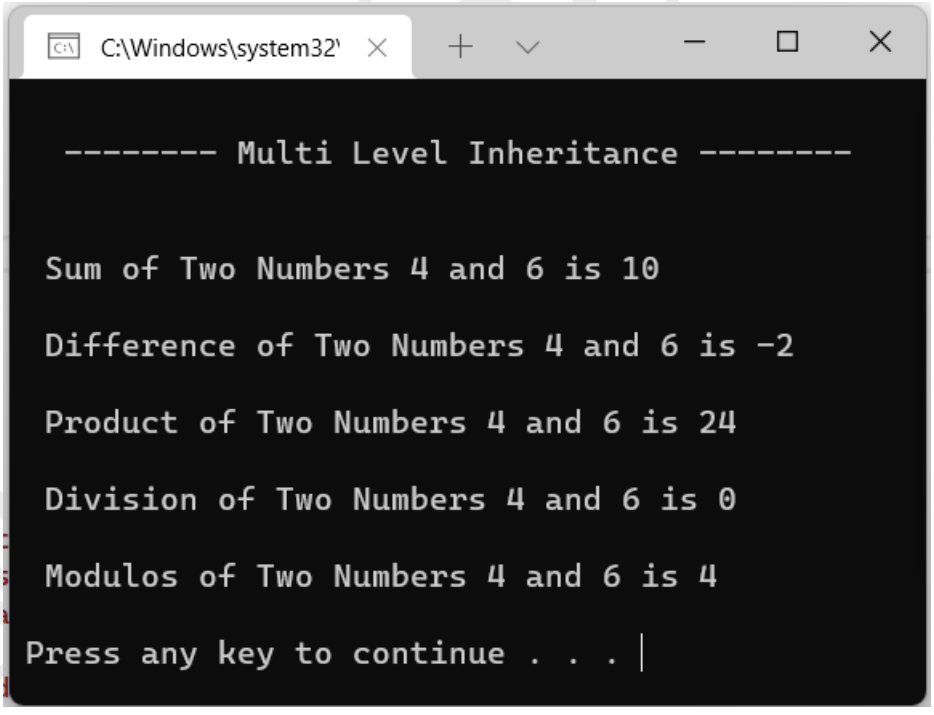
(b). Write Example Code for Multi-Level Inheritance.

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Write a C# Code For Multi-Level Inheritance

// for Reference, check Day 10 Project2 in the same Repository
namespace Day10Project2
{
    class Algebra
    {
        /// <summary>
        /// This is a Addition Method of Two Numbers
        /// </summary>
        /// <param name="a">int A</param>
        /// <param name="b">int B</param>
        /// <returns>Sum</returns>
        public int Add(int a, int b)
        {
            return a + b;
        }
        /// <summary>
        /// This is a Subtraction Method of Two Numbers
        /// </summary>
        /// <param name="a">int A</param>
        /// <param name="b">int B</param>
        /// <returns>Difference</returns>
        public int Sub(int a, int b)
        {
            return a - b;
        }
    }
    /// <summary>
    /// Inheriting the Methods of Parent Class Algebra to Child Class TotalMaths.
    /// </summary>
    class TotalMaths : Algebra
    {
        /// <summary>
        /// This is a Multiplication Method of Two Numbers
        /// </summary>
        /// <param name="a">int A</param>
        /// <param name="b">int B</param>
        /// <returns>Product</returns>
        public int Mul(int a, int b)
        { return a * b; }
    }
    /// <summary>
    /// Inheriting the methods of Parent Class TotalMaths to Child Class OverAllMaths
    /// </summary>
    class OverAllMaths : TotalMaths
    {
        /// <summary>
        /// This is a Division Method of Two Numbers
        /// </summary>
        /// <param name="a">int A</param>
        /// <param name="b">int B</param>
        /// <returns>Division(Quotient)</returns>
        public int Div(int a, int b)
        {
            return a / b;
        }

        /// <summary>
```

```
        /// This is a Modulo-Division Method of Two Numbers
        /// </summary>
        /// <param name="a">int A</param>
        /// <param name="b">int B</param>
        /// <returns>Division(Remainder)</returns>
        public int Mod(int a, int b)
        {
            return a % b;
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            OverAllMaths tm = new OverAllMaths();
            Console.WriteLine("\n  -------- Multi Level Inheritance --------\n");
            Console.WriteLine("\n Sum of Two Numbers 4 and 6 is {0}", tm.Add(4, 6));
            Console.WriteLine("\n Difference of Two Numbers 4 and 6 is {0}", tm.Sub(4, 6));
            Console.WriteLine("\n Product of Two Numbers 4 and 6 is {0}", tm.Mul(4, 6));
            Console.WriteLine("\n Division of Two Numbers 4 and 6 is {0}", tm.Div(4, 6));
            Console.WriteLine("\n Modulos of Two Numbers 4 and 6 is {0}", tm.Mod(4, 6));

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

Output

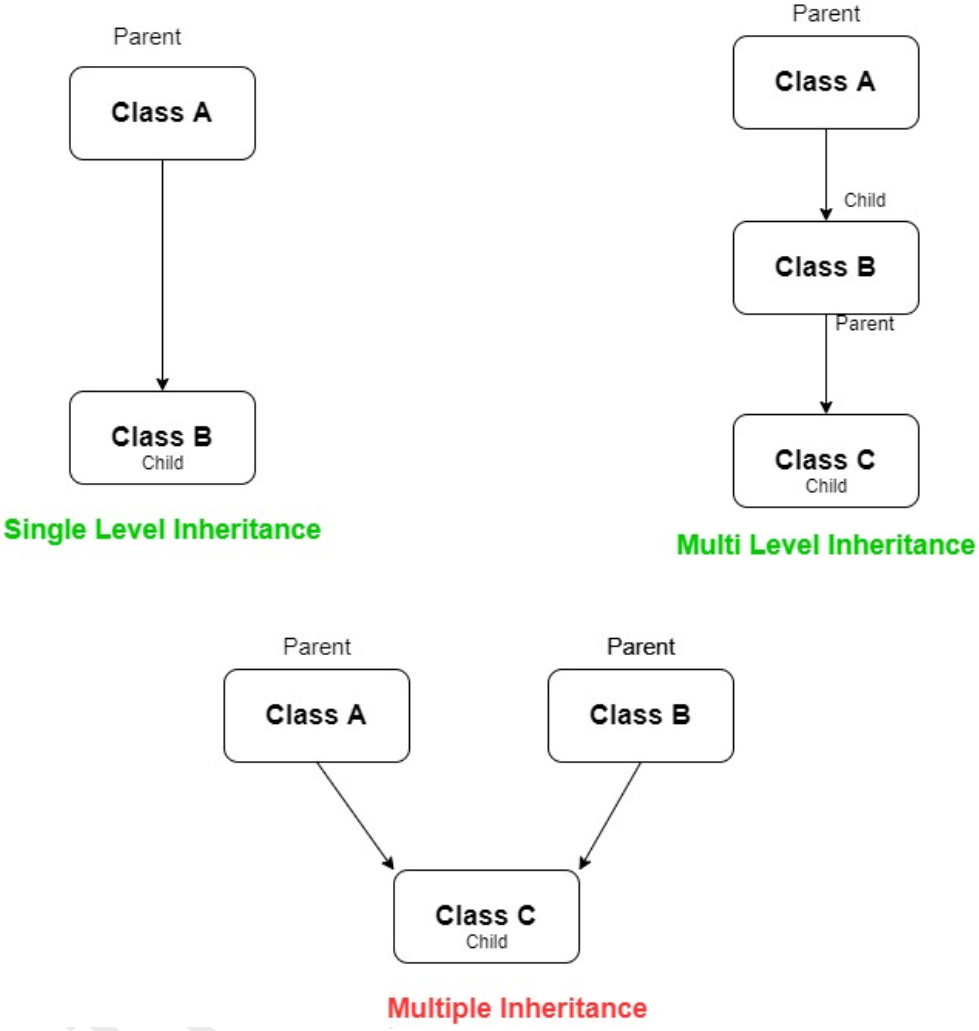


.

Assignment 3

Pictorially represent 3 types of inheritance.

Answer



.

Assignment 4

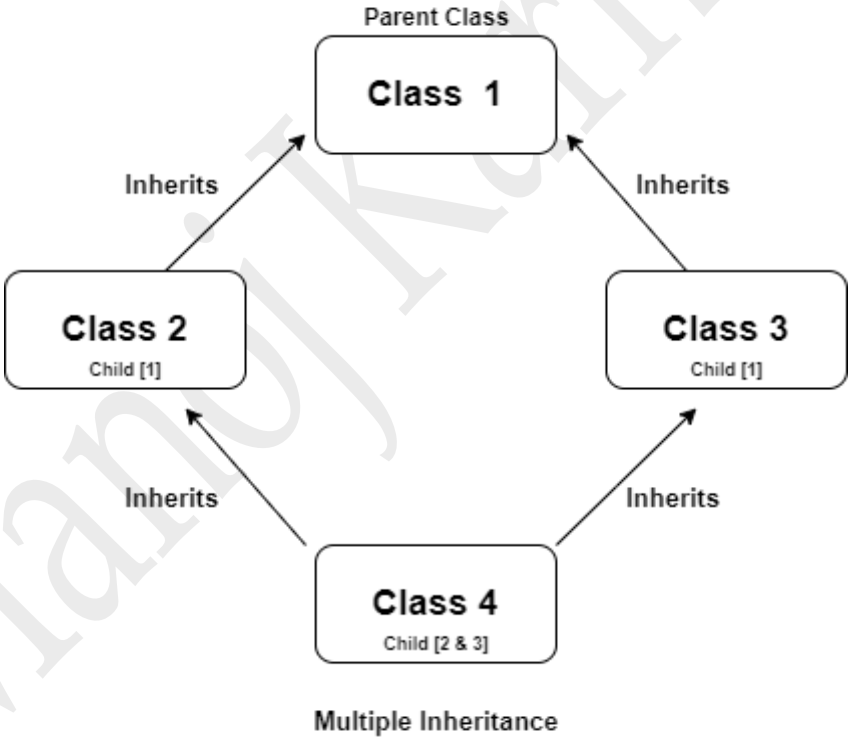
Why Multiple inheritance is not supported for classes in C#?

Answer

C# does not support multiple class inheritance. This is because of 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 clas3 has already overridden the same methods differently.
- So there occurs the **ambiguity** problem wile invoking the methods. In order to solve this problem C# does not support multiple class inheritance.



Multiple Inheritance

.

Assignment 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.

Definition: 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.

In Other words, Polymorphism is the ability of an Object to take on many forms.

Real World Example: One Man Many Forms of Incarnation

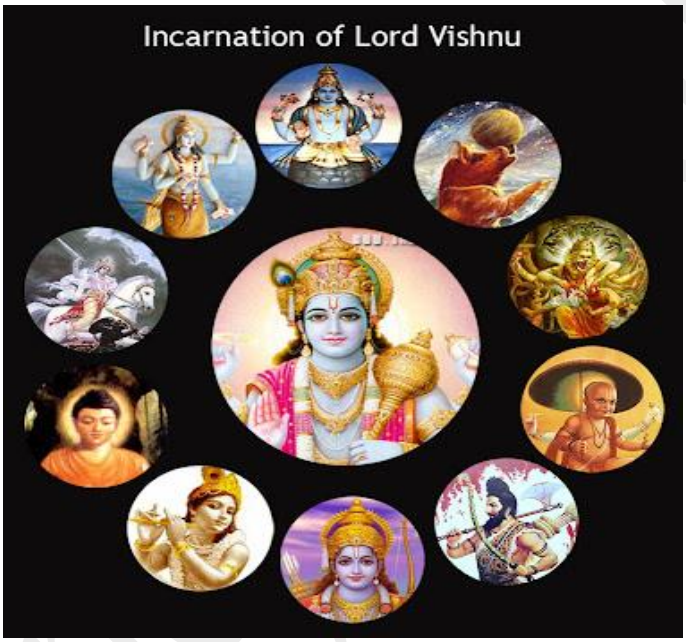
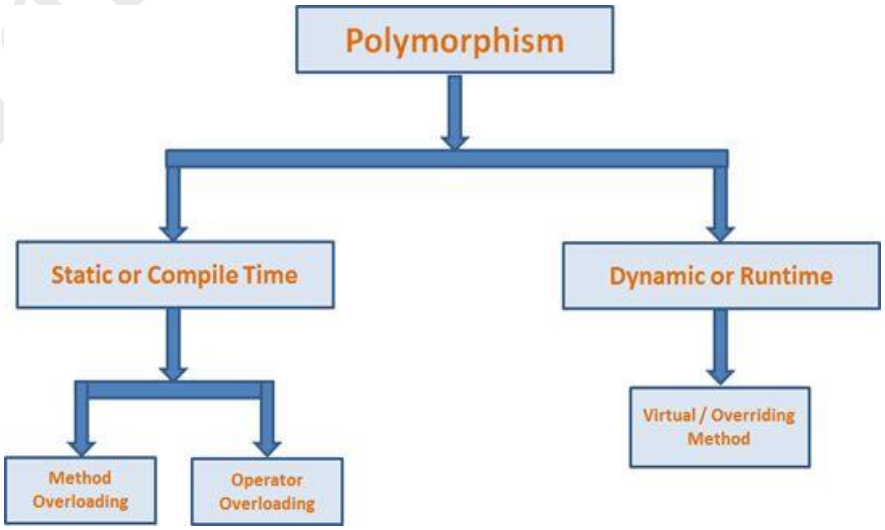


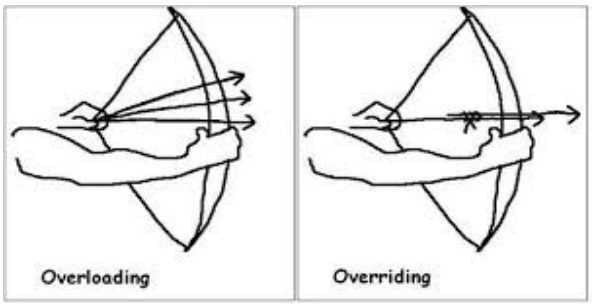
Diagram:



Two Types of Polymorphism:

There two types of Polymorphism in OOPs Concepts. They are:

1. Method Overloading
2. Method Overriding



Assignment 6

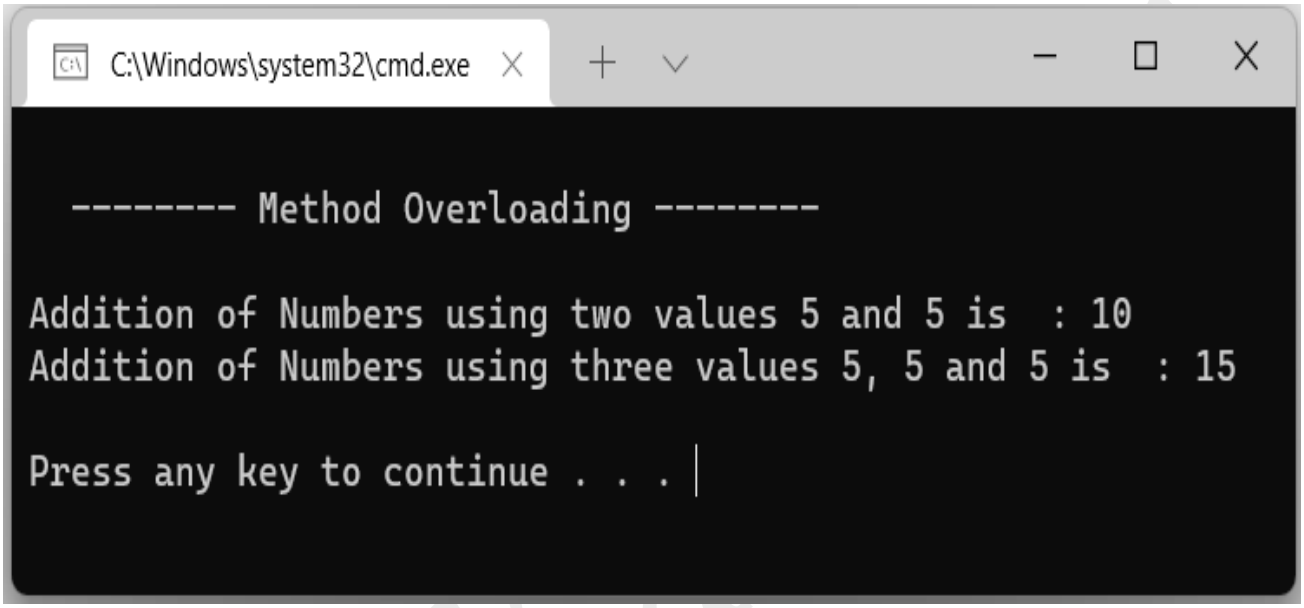
Write Example Code for Method Overloading.

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Write a C# Code for Method Overloading.

// for Reference, check Day10 Project3 in the same Repository.
namespace Day10Project3
{
    class Mathematics
    {
        /// <summary>
        /// This is an Addition Method of Two Numbers
        /// </summary>
        /// <param name="a">Int A</param>
        /// <param name="b">Int B</param>
        /// <returns>Sum</returns>
        public int Add(int a, int b)
        {
            return a + b;
        }
        /// <summary>
        /// This is an Addition Method Of Three Numbers
        /// </summary>
        /// <param name="a">Int A</param>
        /// <param name="b">Int B</param>
        /// <param name="c"></param>
        /// <returns></returns>
        public int Add(int a, int b, int c)
        {
            return a + b + c;
        }

    }
    internal class Program
    {
        static void Main(string[] args)
        {
            Mathematics obj = new Mathematics();
            Console.WriteLine("\n  -------- Method Overloading --------\n");
```

```
            Console.WriteLine("Addition of Numbers using two values 5 and 5 is  :
{0}",obj.Add(5,5));
            Console.WriteLine("Addition of Numbers using three values 5, 5 and 5 is  :
{0}", obj.Add(5,5,5));

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

Output



.

Assignment 7

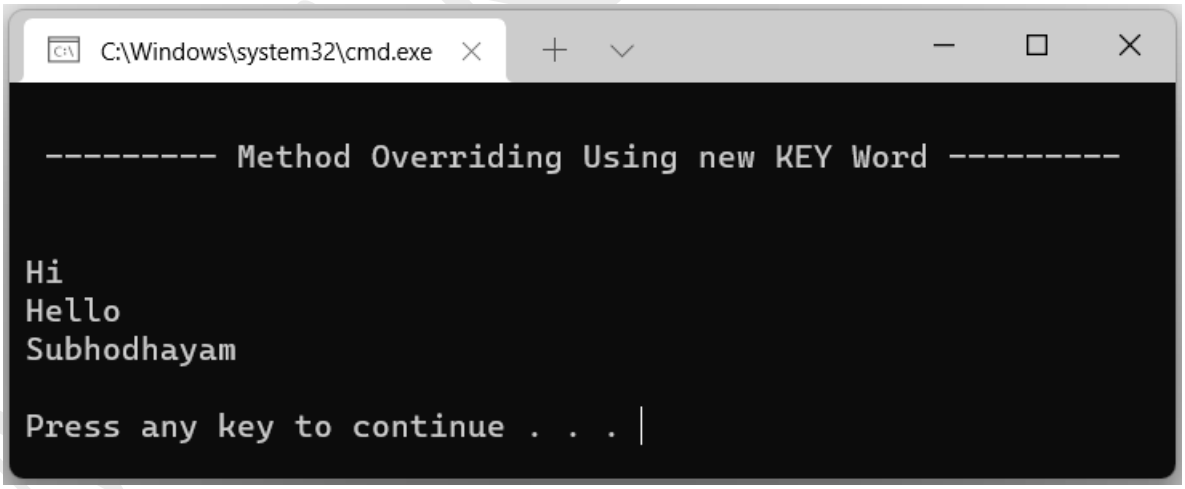
Write Example Code for Method Overriding Using new keyword.

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Write a C# Code for Method Overriding Using new KeyWord.

// for Reference, check Day10Project4 in the same Repository.

namespace Day10Project4
{
    class EnglishGreetings
    {
        /// <summary>
        /// This is a Hi Method, To Greetings Hi
        /// </summary>
        public void PrintHi()
        {
            Console.WriteLine("Hi");
        }
        /// <summary>
        /// This is a Hello Method, To Greetings Hello
        /// </summary>
        public void PrintHello()
        {
            Console.WriteLine("Hello");
        }
```

```
        /// <summary>
        /// This is a Good Morning, To Greeting Good Morning.
        /// </summary>
        public void PrintGM()
        {
            Console.WriteLine("Good Morning");
        }
    }


    class TeluguGreetings : EnglishGreetings
    {
        public new void PrintGM()
        {
            Console.WriteLine("Subhodhayam");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            TeluguGreetings obj = new TeluguGreetings();

            Console.WriteLine("\n ------ Method Overriding Using new KEY Word -------\n");
            obj.PrintHi();
            obj.PrintHello();
            obj.PrintGM();

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

Output



.

Assignment 8

Write Example C# Code for Method Overriding Using virtual, override keyword.

Code

```
using System;

// Authot : Manoj.Karnatapu
// Purpose : Write a C# Code for Method Overriding Using virtual, override keyword.

// For Reference, check Day10Project5 in the same Repository.
namespace Day10Project5
{
    class EnglishGreetings
    {
        /// <summary>
        /// This is a Hi Method, To Greetings Hi
        /// </summary>
        public void PrintHi()
        {
            Console.WriteLine("Hi");
        }
        /// <summary>
        /// This is a Hello Method, To Greetings Hello
        /// </summary>
        public void PrintHello()
        {
            Console.WriteLine("Hello");
        }
        /// <summary>
        /// This is a Good Morning, To Greeting Good Morning.
        /// </summary>
        public virtual void PrintGM()
        {
            Console.WriteLine("Good Morning");
        }
    }


    class TeluguGreetings : EnglishGreetings
    {
        public override void PrintGM()
        {
            Console.WriteLine("Subhodhayam");
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            TeluguGreetings obj = new TeluguGreetings();
            EnglishGreetings obj2 = new EnglishGreetings();

            Console.WriteLine("\n ------ Method Overriding Using virtual , override KEY
Word -------\n");
            obj.PrintHi();
            obj.PrintHello();
            obj2.PrintGM();
            obj.PrintGM();

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

Output