DAY 10 MORNING ASSIGNMENT

(BY G V S S SRI LASYA)

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

- Inheritance means that the derived/child/sub class can inherit and use the methods of base/parent/super class
- Main aim of inheritance is to reuse code and thus avoid unnecessary duplication of code

2) What is polymorphism.

- Same object taking multiple forms is called polymorphism
- In methods,it is of 2 types:

  a)Method overloading:

  If multiple methods within a class share same name but have different number and/or type of parameters,it is called method overloading.Return type of the method doesnt matter here.

  Eg: “WriteLine()” method has a total of 19 forms in C#

  b)Method overriding:

  When a subclass has a method with same method signature (i.e name of the method, number,type and kind (value or reference) of the formal parameters) and same return type as that of a method in its super class then this subclass method is said to override the corresponding method of that super class

3) Why multiple inheritance is not supported for classes in C#

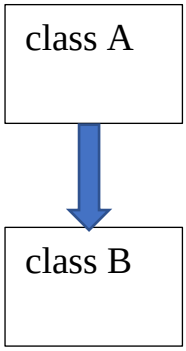
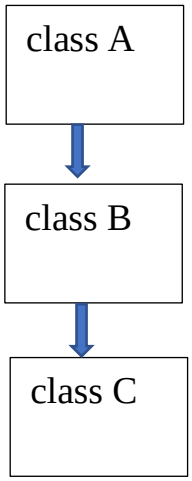
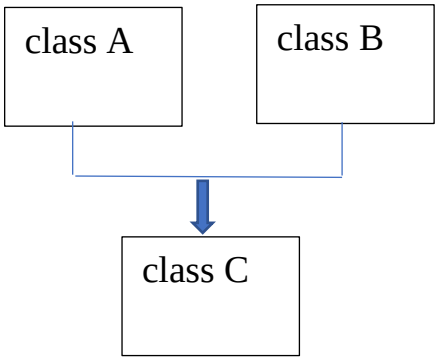
- Multiple inheritance:

It means that a single derived class can inherit from multiple base classes.It is not allowed in C#

- Reason:

Lets take an example where two classes(say class B and class C) are sub classes of same class A and both override a method of class A(say Method()) differently.Another class(say class D) inherits from both class B and class C in multiple inheritance but doesnt override Method().When an object of this class D calls Method() then there would be an ambiguity as to which Method() to call,Method() of class B or of class C.Thus multiple inheritance is not supported for classes in C#.

4) Pictorially represent 3 types of inheritance discussed in the class

Single inheritance	Multi level inheritance	Multiple inheritance
One derived class has only one base class	One base class and one derived class.This derived class acts as a base class to another class	One derived class inherits from multiple base classes
		

5)Write sample code for method overloading

CODE

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


/*************************************************************
Author : G V S S SRI LASYA
Purpose :  Write sample code for method overloading
*************************************************************/


namespace Day10Project1
{
    class Algebra
    {
        /// <summary>
        /// Adds two integers and returns sum
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <returns>
        /// int
        /// </returns>
        public int Add(int a,int b)
        {
            return a + b;
        }


        /// <summary>
        /// Adds 3 integers and returns sum
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <param name="c"></param>
        /// <returns>
        /// int
        /// </returns>
        public int Add(int a, int b,int c)
        {
            return a + b + c;
        }

        /// <summary>
        /// Adds 2 floats and returns sum
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <returns>
        /// float
        /// </returns>
        public float Add(float a,float b)
        {
            return a + b;
        }



    }
    internal class Program
    {
        static void Main(string[] args)
        {
            int a, b;
            float c, d;

            Algebra object1 = new Algebra();

            //taking user inputs
            Console.Write("\nAdding 2 integers");
```

```
            Console.Write("\nEnter first integer : ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter second integer : ");
            b = Convert.ToInt32(Console.ReadLine());

            Console.Write("\n\n\nAdding 2 decimal numbers");
            Console.Write("\nEnter first decimal number : ");
            c = Convert.ToSingle(Console.ReadLine());
            Console.Write("Enter second decimal number: ");
            d = Convert.ToSingle(Console.ReadLine());


            //printing output
            Console.Write($"\n\n\nSum of the 2 integers is : {object1.Add(a,
b)}");
            Console.Write($"\n\nSum of the 2 decimal numbers is :
{object1.Add(c,d)}");

            Console.ReadLine();







        }
    }
}
```

OUTPUT

```
Adding 2 integers
Enter first integer : 45
Enter second integer : 34



Adding 2 decimal numbers
Enter first decimal number : 45.66
Enter second decimal number: 35.23



Sum of the 2 integers is : 79

Sum of the 2 decimal numbers is : 80.89
```

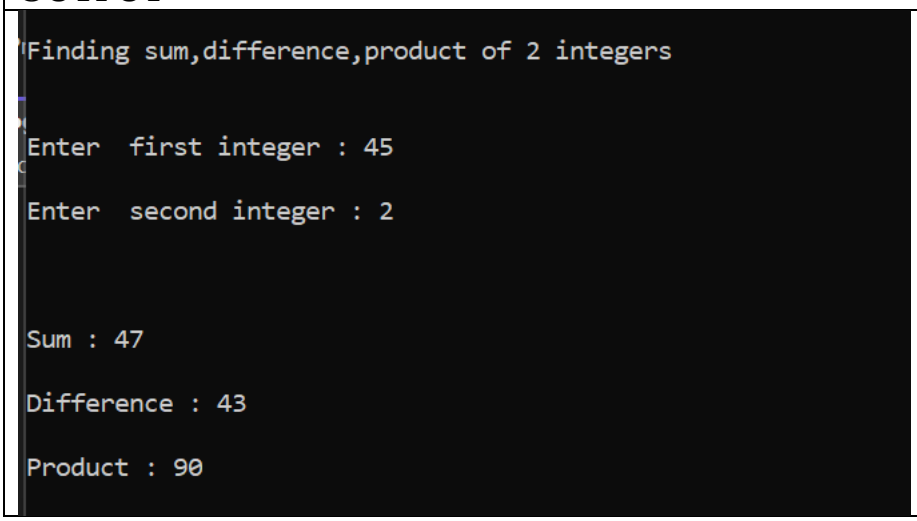
6) Write example code for single inheritanc

CODE

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

/*****************************************************************
Author : G V S S SRI LASYA
Purpose : Write example code for single inheritance
*****************************************************************/


namespace day10project2
{
    class MathsOperaations1
    {
        /// <summary>
        /// adds 2 integers and retuns sum
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <returns>
        /// int
        /// </returns>
        public int Add(int a, int b)
        {
            return a + b;
        }

        /// <summary>
        /// subtracts 2 integers and returns difference
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <returns>
        /// int
        /// </returns>
        public int Subtract(int a,int b)
        {
            return a - b;
        }

    }

    class MathsOperations2 : MathsOperaations1
    {
        /// <summary>
        /// multiplies 2 integers and returns product
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <returns>
        /// int
        /// </returns>
        public int Multiply(int a,int b)
        {
            return a * b;
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            int a, b,c=0,d=0,e=1;

            MathsOperations2 object1 = new MathsOperations2();

            Console.Write("\nFinding sum,difference,product of 2 integers");

            //taking user inputs
            Console.Write("\n\n\nEnter  first integer : ");
```

```
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("\nEnter  second integer : ");
            b = Convert.ToInt32(Console.ReadLine());

            //calling methods using object
            c = object1.Add(a, b);
            d = object1.Subtract(a, b);
            e = object1.Multiply(a, b);

            //printing outputs
            Console.Write($"\n\n\nSum : {c}");
            Console.Write($"\n\nDifference : {d}");
            Console.Write($"\n\nProduct : {e}");

            Console.ReadLine();

        }
    }
}
```

OUTPUT



7) Write example code for multi level inheritance

CODE

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

/*****************************************************************
Author : G V S S SRI LASYA
Purpose : Write example code for multi level inheritance
*****************************************************************/


namespace day10project2
{
    class MathsOperaations1
    {
        /// <summary>
        /// adds 2 integers and retuns sum
        /// </summary>
        /// <returns>
        /// int
        /// </returns>
        public int Add(int a, int b)
        {
            return a + b;
        }

        /// <summary>
        /// subtracts 2 integers and returns difference
        /// </summary>
        /// <returns>
        /// int
        /// </returns>
        public int Subtract(int a, int b)
        {
            return a - b;
        }

    }

    class MathsOperations2 : MathsOperaations1
    {
        /// <summary>
        /// multiplies 2 integers and returns product
        /// </summary>
       /// <returns>
        /// int
        /// </returns>
        public int Multiply(int a, int b)
        {
            return a * b;
        }
    }

    class MathsOperations3 : MathsOperations2
    {
        /// <summary>
        /// Divides 2 integers and returns quotient
        /// </summary>
        /// <returns>
        /// float
        /// </returns>
        public float Divide(int a, int b)
        {
            return (float)a / b;
        }
    }


    internal class Program
    {
```

```
        static void Main(string[] args)
        {
            int a, b, c = 0, d = 0, e = 1;
            float f = 1f;

            MathsOperations3 object1 = new MathsOperations3();

            Console.Write("\nFinding sum,difference,product,quotient of division for 2
integers");

            //taking user inputs
            Console.Write("\n\n\nEnter  first integer : ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("\nEnter  second integer : ");
            b = Convert.ToInt32(Console.ReadLine());

            //calling methods using object
            c = object1.Add(a, b);
            d = object1.Subtract(a, b);
            e = object1.Multiply(a, b);
            f = object1.Divide(a, b);

            //printing outputs
            Console.Write($"\n\n\nSum : {c}");
            Console.Write($"\n\nDifference : {d}");
            Console.Write($"\n\nProduct : {e}");
            Console.Write($"\n\nQuotient of division : {f}");

            Console.ReadLine();

        }
    }
}
```

OUTPUT

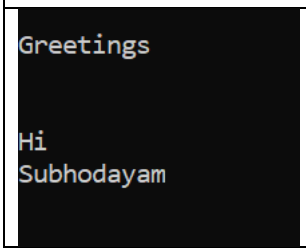
```
Finding sum,difference,product,quotient of division for 2 integers


Enter  first integer : 33

Enter  second integer : 2



Sum : 35

Difference : 31

Product : 66

Quotient of division : 16.5
```

8) 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;

/*************************************************************************
Author : G V S S SRI LASYA
Purpose :  Write sample code for method overriding(using new key word)
*************************************************************************/


namespace day10Project4
{
    class Greeting
    {
        /// <summary>
        /// Prints "Hi"
        /// </summary>
        ///<return>
        ///void
        /// </return>
        public void PrintHi()
        {
            Console.Write("\nHi");
        }


        /// <summary>
        /// Prints Good morning message in English
        /// </summary>
        /// <return>
        /// void
        /// </return>
        public void PrintGoodMornig()
        {
            Console.Write("\nGood morning");

        }

    }

    class GreetingInTelugu : Greeting
    {
        /// <summary>
        /// Prints Good morning in Telugu
        /// </summary>
        /// <return>
        /// void
        /// </return>
        public new void PrintGoodMornig()
        {
            Console.Write("\nSubhodayam");

        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
            GreetingInTelugu object1 = new GreetingInTelugu();

            Console.Write("\nGreetings\n\n");
            object1.PrintHi();
            object1.PrintGoodMornig();


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

OUTPUT



9) Write sample code for method overloading

CODE

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


/*************************************************************
Author : G V S S SRI LASYA
Purpose :  Write sample code for method overloading
*************************************************************/


namespace Day10Project1
{
    class Algebra
    {
        /// <summary>
        /// Adds two integers and returns sum
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <returns>
        /// int
        /// </returns>
        public int Add(int a,int b)
        {
            return a + b;
        }


        /// <summary>
        /// Adds 3 integers and returns sum
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <param name="c"></param>
        /// <returns>
        /// int
        /// </returns>
        public int Add(int a, int b,int c)
        {
            return a + b + c;
        }

        /// <summary>
        /// Adds 2 floats and returns sum
        /// </summary>
        /// <param name="a"></param>
        /// <param name="b"></param>
        /// <returns>
        /// float
        /// </returns>
        public float Add(float a,float b)
        {
            return a + b;
        }



    }
    internal class Program
    {
        static void Main(string[] args)
        {
            int a, b;
            float c, d;

            Algebra object1 = new Algebra();

            //taking user inputs
```

```
            Console.Write("\nAdding 2 integers");
            Console.Write("\nEnter first integer : ");
            a = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter second integer : ");
            b = Convert.ToInt32(Console.ReadLine());

            Console.Write("\n\n\nAdding 2 decimal numbers");
            Console.Write("\nEnter first decimal number : ");
            c = Convert.ToSingle(Console.ReadLine());
            Console.Write("Enter second decimal number: ");
            d = Convert.ToSingle(Console.ReadLine());


            //printing output
            Console.Write($"\n\n\nSum of the 2 integers is : {object1.Add(a, b)}");
            Console.Write($"\n\nSum of the 2 decimal numbers is : {object1.Add(c,d)}");

            Console.ReadLine();




        }
    }
}
```

OUTPUT