DAY 11 MORINING ASSIGNMENT

DATE:07/02/2022

DAY:MONDAY

- Write the six points about interface discussed in the class

1. Interface is pure abstract class
2. Interface name should start with I
3. Interface acts like a contract
4. By default the methods in interface are public and abstract
5. Any class that is implementing interface must overrides all the Methods

Interface support multiple interface

- Write the seven points discuss about properties

1.properties are almost same as class variables with get;and set;
2.A property with only get is readonly
3.Aproperty with only set is writeonly
4.A property with get andset => you can read value and assign the value
5.properties are introduced to deal with private variable
6.A very simple example properties
7. properties names start with upper case

- Research write the difference between abstract class and interface in c#

INTERFACE	ABSTARCT
1.Interface supports multiple inheritance	1.Abstarct class does not support multiple inheritance
2.interface does'n contains constructors	2.Abstarct class contains constructors
3.member of interface can not be static	3.only complete member of abstract class can be static

- Write th e example program for interfaces discussed in the class IShape,include the class circle,square,triangle,rectangle

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

namespace Day_11_project
{


        interface IShape
        {
            int CalculatePerimeter();
            int CalculateArea();

    }

        class Circle : IShape
        {
            int radius;

            public void ReadRadius()
            {
                Console.WriteLine("Enter Radius");
                radius = Convert.ToInt32(Console.ReadLine());
            }

            public int CalculateArea()
            {
                return 22 * radius * radius / 7;
            }
            public int CalculatePerimeter()
            {
                return 2 * 22 * radius / 7;
            }
        }
        class Square : IShape
        {
            private int side;

            public void Readdata()
            {
                Console.WriteLine("Enter Side");
                side = Convert.ToInt32(Console.ReadLine());
            }

            public int CalculateArea()
            {

                return side * side;
            }
            public int CalculatePerimeter()
            {
                return 4 * side;
            }
        }
```

```
        class Triangle : IShape
        {
            private int x;
            private int y;
            private int z;

            public void ReadSide()
            {
                Console.WriteLine("Enter Side");
                x = Convert.ToInt32(Console.ReadLine());
                y = Convert.ToInt32(Console.ReadLine());
                z = Convert.ToInt32(Console.ReadLine());
            }

            public int CalculateArea()
            {

                return x * y * z;
            }
            public int CalculatePerimeter()
            {

                return x + y + z;
            }
        }
        class Rectangle : IShape
        {
            private int length;
            private int breadth;


            public void ReadSide()
            {
                Console.WriteLine("Enter Side");
                length = Convert.ToInt32(Console.ReadLine());
                breadth = Convert.ToInt32(Console.ReadLine());
            }

            public int CalculateArea()
            {

                return length * breadth;
            }
            public int CalculatePerimeter()
            {
                return 2 * (length + breadth);
            }
        }
        internal class program
        {
            static void Main(String[] args)
            {
                Circle c = new Circle();
                c.ReadRadius();
                Console.WriteLine(c.CalculatePerimeter());
                Console.WriteLine(c.CalculateArea());
                Square s = new Square();
                s.Readdata();
                Console.WriteLine(s.CalculatePerimeter());
                Console.WriteLine(s.CalculateArea());
                Triangle t = new Triangle();
                t.ReadSide();
                Console.WriteLine(t.CalculatePerimeter());
```

```
                Console.WriteLine(t.CalculateArea());
                Rectangle r = new Rectangle();
                r.ReadSide();
                Console.WriteLine(r.CalculatePerimeter());
                Console.WriteLine(r.CalculateArea());

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

OUTPUT