Assingment_14

2022

FEBRUARY 11

@NB_Healthcare.tech
Done by: N. Prudhvi

NB Healthcare Technologies

Q. WACP to illustrate sealed class.

CODE

```
sealed class Hospital
    {

  public void Ambulance()
        {
            Console.WriteLine("DIAL 108 for ambulance");
        }

    }
    internal class Program
    {
            /****************************************
             * Author : Prudhvi                      *
             * Purpose: To declare a sealed class.   *
             * ****************************************/
        static void Main(string[] args)
        {
            var obj1 = new Hospital();
            obj1.Ambulance();
            Console.ReadLine();
        }
    }
```

OUTPUT

```
C:\Windows\system32\cmd.exe
DIAL 108 for ambulance
```

Q. WACP to illustrate normal properties & auto implementation.

CODE

```
//Normal Properties
        private float current;
        private float voltage;
        public float Voltage
        {
            set
            {
                voltage = value;
            }
        }
        public float Current
        {
            set
            {
                current = value;
            }
        }
        public float Power
        {
            get
            {
                return current * voltage;
```

```
            }
        }
        //Auto - Implemented Properties
        public float Resistance
        {
            get
            {
                return voltage * current/current*current;
            }
        }
    }
    internal class Program
    {
        static void Main(string[] args)
        {
     /*************************************************************************
      * Author : Prudhvi                                                      *
      * Purpose: To declare a sealed class.                                   *
      *************************************************************************/


            Console.Write("Enter amps = ");
            float a = float.Parse(Console.ReadLine());
            var instance = new TotalPower();
            instance.Current = a;
            Console.WriteLine();
            Console.Write("Enter voltage =");
            float b = float.Parse(Console.ReadLine());
            instance.Voltage = b;
            Console.WriteLine();
            Console.WriteLine($"watts = {instance.Power}");
            Console.WriteLine();
            Console.WriteLine($"watts = {instance.Resistance}");
            Console.ReadLine();
        }
    }
```

OUTPUT

```
C:\Windows\system32\cmd.exe
Enter amps = 5

Enter voltage =230

Power = 1150watts

Resistance = 46ohms
```

Q. Difference b/w normal properties auto implementation.

A property is a member that allows you to read, write, or compute the value of a private field in a flexible way. Properties are special procedures known as accessor that can be utilized as if they were public data members. This allows data to be easily accessed while also aiding in the promotion of the product. Methods' safety and flexibility

Auto-Implemented Properties: Auto-implemented properties are available in C# 3.0 and later. When no additional logic is necessary in the property accessor, the declaration becomes more concise. They also make it possible to create objects, you'll need client code.

Q.Prime check

CODE

```
public class Num_check
    {
        public int Num_br;
        public bool IsPrime(int a,int b =3)
        {
            bool flag = false;
            if (a<=1)
                Console.WriteLine("Enter number unsigned numbers more than one");
            else
            {
                while (b < a )
                {
                    if (a % b == 0)
                    {
                        flag = true;
                        break;
                    }
                    b++;
                }
            }
            Num_br = b;
            return flag;

        }
    }

    internal class Program
    {


        static void Main(string[] args)
        {
            /****************************************
             * Author : Prudhvi                     *
             * Purpose: To Prime Check.             *
             * ************************************/
```

```
        static void Main(string[] args)

            var PNC = new Num_check();
            Console.Write("Enter your value:");
            int prime_num = Convert.ToInt32(Console.ReadLine());

            if (PNC.IsPrime(prime_num) == true)
            {
                Console.WriteLine($"{prime_num} it is composite number {PNC.Num_br} ");
            }
            else
            {
                Console.WriteLine($"{prime_num} is a prime number");
            }
            Console.ReadLine();
        }
    }
```

OUTPUT

```
C:\Windows\system32\cmd.exe
Enter your value:734432
734432 it is composite number 4
```

Q. Print 1to30 num_s except multipels of 3

CODE

```
static void Main(string[] args)
        {
            int I = 1;
            while (I<=30)
            {
                if (I % 3 == 0)
                {

                    I++;
                    continue;

                }
                Console.WriteLine(I);
                I++;
            }

        }
```

OUTPUT

```
C:\Windows\system32\cmd.exe
1,2,4,5,7,8,10,11,13,14,16,17,19,20,22,23,25,26,28,29,
```

Q. 1st Number divided by 97 after 1000.

CODE

```
static void Main(string[] args)
        {
            int Num = 1000;
            int mulp = 97;
            while(Num < Num+mulp)
            {
                if (Num % mulp == 0)
                {
                    Console.WriteLine("1st Number diivde by 97 after 1000 ="+Num+".");
                    break;
                }
                Num++;
            }
        }
```

OUTPUT

C:\Windows\system32\cmd.exe

```
1st Number diivde by 97 after 1000 =1067.
```

Q.