Day 14 Assignments

By

Praveen Chakravarthi

10-02-2022

NB Health Care

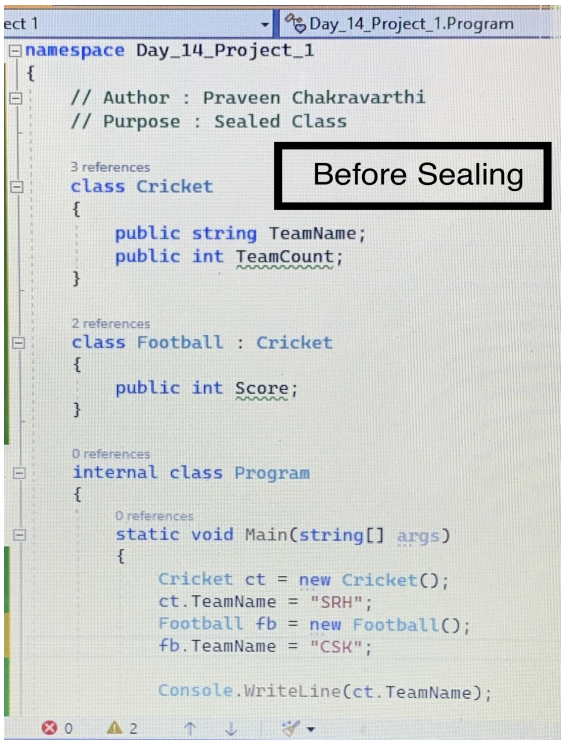
1. Research and write what is the use of sealed class, WACP to illustrate sealed class.

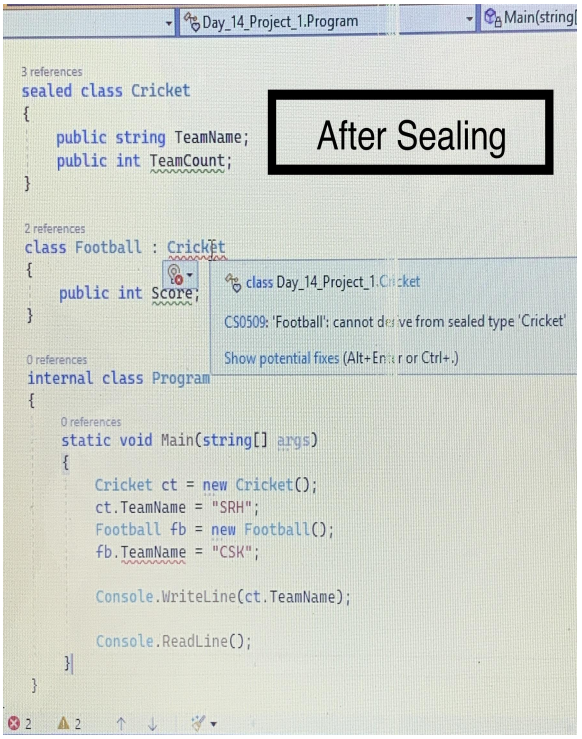
- **Sealed Class** is used to stop the class to be inherited by other classes.
- **”Sealed”** is the Keyword used to seal the class.

Code :

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


namespace Day_14_Project_1
{
    // Author : Praveen Chakravarthi
    // Purpose : Sealed Class

    sealed class Cricket
    {
        public string TeamName;
        public int TeamCount;
    }

    class Football : Cricket
    {
        public int Score;
    }

    internal class Program
    {

        static void Main(string[] args)
        {

            Cricket ct = new Cricket();
            ct.TeamName = "SRH";
            Football fb = new Football();
            fb.TeamName = "CSK";

            Console.WriteLine(ct.TeamName);

            Console.ReadLine();

        }
    }
}
```

Output :





2. Research and write what is the difference between normal properties and auto-implemented properties,

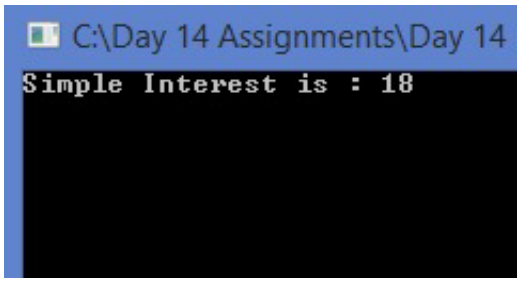
Normal Properties	Auto-Implemented Properties
• Normal Properties have pre declared variables and can access them	• Auto-Implemented Properties have instance variables
• Normal Properties can have getter or setter or both	• Auto-Implemented Properties must have getter
• Normal Properties can have ReadOnly or WriteOnly	• Auto-Implemented must have ReadOnly

Code :WACP to illustrate normal properties

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

namespace Day_14_Project_2
{
    // Author : Praveen Chakravarthi
    // Purpose : Normal Properties

    class SimpleInterest
    {
        private int principle;
        private int time;
        private int rate;
        private int si;

        public int Principle
        {
            set
            {
                principle = value;
            }
        }
        public int Time
        {
            set
            {
                time = value;
            }
        }
        public int Rate
        {
            set
            {
                rate = value;
            }
        }
        public int SI
        {
            get
            {
                si = (principle*time*rate)/ 100;
                return si;
            }
        }

    }

    internal class Program
    {
```

```
        static void Main(string[] args)
        {
            SimpleInterest sit = new SimpleInterest();
            sit.Principle = 20;
            sit.Time = 3;
            sit.Rate = 30;

            Console.WriteLine("Simple Interest is : {0}",sit.SI);
            Console.ReadLine();
        }
    }
}
```

Output :

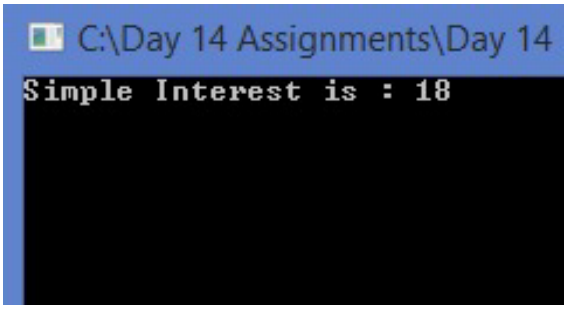


Code :

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

namespace Day_14_Project_3
{
    // Author : Praveen Chakravarthi
    // Purpose : Auto-implemented Properties
    internal class Program
    {
        class SimpleInterest
        {
            private int principle;
            private int time;
            private int rate;

            public int Principle
            {
                set
                {
                    principle = value;
                }
            }
            public int Time
            {
```

```
            set
            {
                time = value;
            }
        }
        public int Rate
        {
            set
            {
                rate = value;
            }
        }

        // Auto-implmented Property
        public int SI
        {
            get
            {
              return  (principle * time * rate) / 100;

            }
        }

    }
    static void Main(string[] args)
    {
        SimpleInterest sit = new SimpleInterest();
        sit.Principle = 20;
        sit.Time = 3;
        sit.Rate = 30;

        Console.WriteLine($"Simple Interest is : {sit.SI}");
        Console.ReadLine();
    }
  }
}
```

Output :

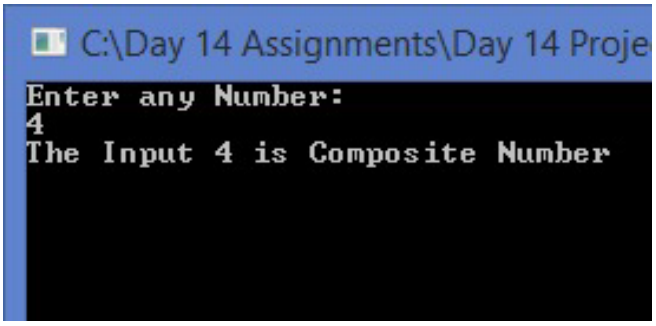


4. WACP to check if the number is prime or not using logic discussed in the class HINT : use break;

Code :

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

namespace Day_14_Project_5
{
    // Author : Praveen Chakravarthi
    // Purpose : Prime Number program using Break
    internal class Program
    {
        static void Main(string[] args)
        {
            int i, n;
            Console.WriteLine("Ënter any Number: ");
            n = Convert.ToInt32(Console.ReadLine());

            for ( i = 2; i < n; i++)
            {
                if (n % i == 0)
                    break;
            }
                if (i == n)

                    Console.WriteLine("The Input {0} is Prime Number", n);
                else
                    Console.WriteLine("The Input {0} is Composite Number", n);
            Console.ReadLine();
        }
    }
}
```

Output :

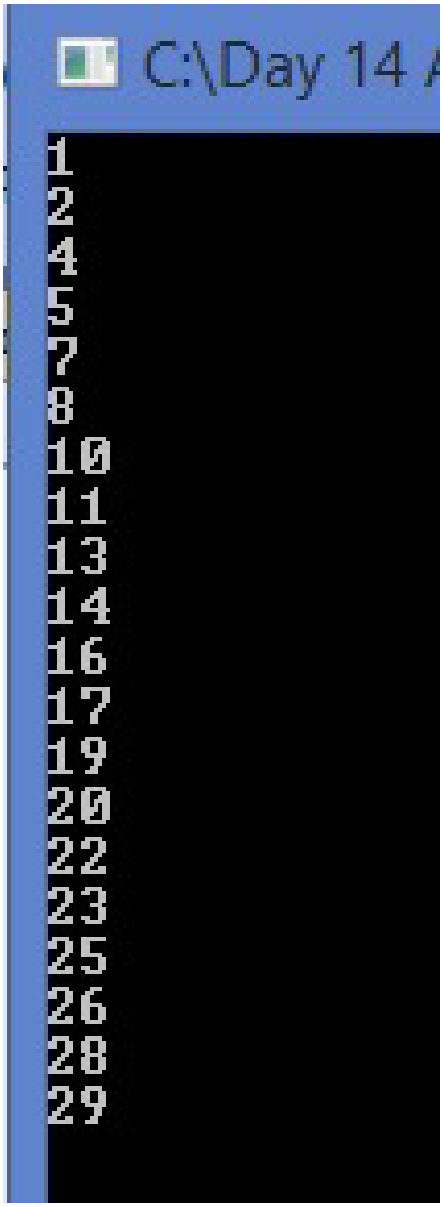


5. print numbers from 1 to 30 and skip the numbers divisible by 3
HINT: use continue;

Code :

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

namespace Day_14_Project_6
{
    // Author : Praveen Chakravarthi
    // Purpose : Printing Numbers between 1 & 30 and skip numbers divisible by 3 using
continue
    internal class Program
    {
        static void Main(string[] args)
        {
            int i;

            for (i = 1; i <30; i++)
            {
                if (i % 3 == 0)
                    continue;
                Console.WriteLine(i);
            }

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

Output :



6. Find the first number after 1000 which is divisible by 97 **HINT: use for loop and break**
Code :
using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading.Tasks;

```
namespace Day_14_Project_7
{
    // Author : Praveen Chakravarthi
    // Purpose : Print number which is divisible by 97 just after 1000
    internal class Program
    {
        static void Main(string[] args)
        {
            int n = 97;
            int x;
            for (x=1000; x < 1097; x++)
            {
                if (x % n == 0)
                    break;
            }
            Console.WriteLine($"The number divisible by 97 just after 1000 is {x}");
            Console.ReadLine();
        }
    }
}
```

Output :