Day 14 Assignment
By
Triveni Anumolu
10-02-2022

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

Sealed class:

- A sealed class is same as a normal class. It has methods, variables, properties.
- A sealed class prevents other classes from inheriting from it.

Code:

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

namespace Day14project1
{
   sealed class Sample
   {
      public static string s = "Hi";
      public string walk()
      {
         return "Go for a walk";
      }
   }
   class Program
   {
      static void Main(string[] args)
      {
         Sample s1 = new Sample();
         Console.WriteLine(Sample.s);
         Console.WriteLine(s1.walk());

         Console.ReadLine();

      }
   }
}
```

Result:

```
Hi
Go for a walk
```

2. Research and write what is the difference between normal properties and auto-implemented properties.
WACP to illustrate normal properties.
WACP to illustrate auto-implemented properties

Difference between normal and auto implemented properties:

Normal Properties	Auto-implemented properties
1.Normal properties point to private variables.	1.Auto-implemented properties do not point to any variables.
2.set accessor is used to assign a value to variable. get returns the value of variable.	2.get accessor is must in auto-implemented properties.

Code for normal properties:

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

namespace Day14Project5
{
   class Calculation
   {
      /*********************************************
       * Author: Triveni Anumolu
       * Purpose:Finding force using normal properties
       * *********************************************/
      private int mass;
      private int acceleration;

      private int force;
      //Normal Properties
      public int Mass
      {
         set
         {
            mass = value;
         }
      }
      public int Acceleration
      {
         set
         {
            acceleration = value;
         }
      }
```

```
        public int Force
        {
            get
            {
                return force = mass * acceleration;
            }
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Calculation cal = new Calculation();
            cal.Mass = 100;
            cal.Acceleration = 20;
            Console.WriteLine(cal.Force);
            Console.ReadLine();
        }
    }
}
```

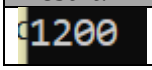
Result:

Code for auto-implemented property:

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

namespace Day14Project6
{
    /*************************************************
     * Author:Triveni Anumolu
     * Purpose:Finding force using auto-implemented property
     * ***********************************************/
    class Calculation
    {
        private int mass;
        private int acceleration;
        private int force;
        public int Force
        {
            get
            {
```

```
            return force = mass * acceleration;
        }
    }

    static void Main(string[] args)

        Calculation cal = new Calculation();
        cal.mass = 12;
        cal.acceleration = 100;
        //Console.WriteLine(cal.Force);
        Console.ReadLine();
    }
  }
}
```

Result:



4. WACP to check if the number is prime or not by 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 Day14project2
{
  class Program
  {
    /*****************************************************************
     Author: Triveni Anumolu
     Purpose:Finding whether a number is prime or not by using break
    *****************************************************************/
    static void Main(string[] args)
    {
        int i, n = 99;
        for( i=2;i<n;i++)
        {
            if (n % i == 0)
                break;
        }
```

```
        if(i==n)
           Console.WriteLine("{0} is a prime number",n);
        else
           Console.WriteLine("{0} is not a prime number",n);
        Console.ReadLine();

      }
   }
}
```

Result:

```
99 is not a prime number
```

5. WACP to print numbers from 1 to 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 Day14project3
{
   class Program
   {

/*************************************************************************
        Author:Triveni Anumolu
      Purpose:Printing the values which are divisible by 3 using continue

*************************************************************************/

      static void Main(string[] args)
      {
         int n = 30;
         for (int i = 1; i <= n; i++)
         {
            if (i % 3 == 0)
               continue;
            Console.Write(i);

         }
```

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

Result:

```
1 2 4 5 7 8 10 11 13 14 16 17 19 20 22 23 25 26 28 29
```

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 Day14project4
{
    class Program
    {
        /*************************************************************
         Author:Triveni Anumolu
         Purpose:Finiding the first number after 1000 which is divisible by 97 using for loop and
                    break
        *************************************************************/
        static void Main(string[] args)
        {
            int n = 97;
            for(int i=1000;i<=1097;i++)
            {
                if (i % n == 0)
                {
                    Console.WriteLine(i);
                    break;
                }

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

Result:

```
1067
```