DAY 14 Assignment
By
Nanam Vaishnavi
10 - Feb - 2022

1) Research and write what is the use of sealed class.
Write a C# program to illustrate sealed class.

Sealed Class

- It is a class that can't be inherited by another class but can be instantiated.
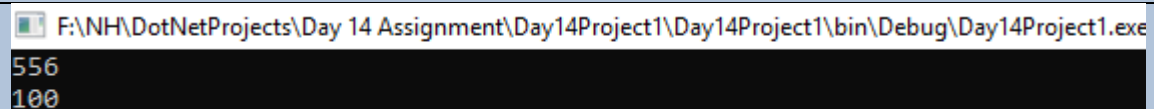- It can be used as parent class or base class.

CODE :

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// ********************************************************
// Author : Nanam Vaishnavi
// Purpose : Sealed class
// ********************************************************

namespace Day14Project1
{
    sealed class Police
    {
        public static int Helpline = 100;

        public string Getsecret()
        {
            return "556";
        }
    }


    internal class Program
    {
        static void Main(string[] args)
        {
            Police p = new Police();
            Console.WriteLine(p.Getsecret());
            Console.WriteLine(Police.Helpline);
            Console.ReadLine();
        }
    }
}
```

OUTPUT

F:\NH\DotNetProjects\Day 14 Assignment\Day14Project1\Day14Project1\bin\Debug\Day14Project1.exe

```
556
100
```

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.

Normal Properties	Auto – Implemented Properties
• It is a member that provides a flexible mechanism for classes to expose private fields.	• It enables you to quickly specify a property of a class without having to write code to get and set the property.
• These are similar to class variables with **get; & set;** Methods.	• It must consists of **get;** method and set; is optional.

Write a C# program to illustrate normal properties

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// ***************************************************
// Author : Nanam Vaishnavi
// Purpose : To illustrate normal properties
// ***************************************************

namespace Day14Project_2
{
    public class Employee
    {
        private int id;
        public string name;
        public int  salary;

        public int Id
        {
            get
            {
                return id;
            }
            set
            {
                id = value;
            }
        }
        public string Name
        {
            get
            {
                return name;
            }
            set
            {
                name = value;
            }
        }
        public int Salary
        {
            Get
            {
```

```csharp
                return salary;
            }
            set
            {
                salary = value;
            }
        }
    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Employee emp = new Employee();
            emp.Id = 501;
            emp.Name = "Vaishnavi";
            emp.Salary = 30000;
            Console.Write($"id={emp.Id}, name={emp.Name}, salary=
{emp.Salary}");

            Console.ReadLine();
        }

    }
}
```

OUTPUT

Select F:\NH\DotNetProjects\Day 14 Assignment\Day14Project_2\Day14Project_2\bin\Debug\Day14Proj
id=501, name=Vaishnavi, salary= 30000

Write a C# Program to illustrate auto-implemented properties

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// ****************************************************
// Author : Nanam Vaishnavi
// Purpose : Auto – implemented property
// ****************************************************

namespace Day14Project3
{
    class Customer
    {
        private string name {get; set;}
        private int id {get; set;}
        private string email {get; set;}

        public void SetId(int Id)
        {
            this.id = Id;
        }
        public int GetId()
        {
            return this.id;
        }
    }
```

```csharp
        public void SetName(string Name)
        {
            this.name = Name;
        }
        public string GetName()
        {
            return this.name;
        }

        public void SetEmail(string Email)
        {
            this.email =Email;
        }
        public string GetEmail()
        {
            return this.email;
        }


    }

    internal class Program
    {
        static void Main(string[] args)
        {
            Customer c = new Customer();
            c.SetId(501);
            c.SetName("Vaishnavi");
            c.SetEmail("abcd@gmail.com");
            Console.WriteLine(c.GetId());
            Console.WriteLine(c.GetName());
            Console.WriteLine(c.GetEmail());


            Console.ReadLine();

        }
    }
}
```

OUTPUT

F:\NH\DotNetProjects\Day 14 Assignment\Day14Project3\Day14Project3\bin\Debug\Day14P

```
501
Vaishnavi
abcd@gmail.com
```

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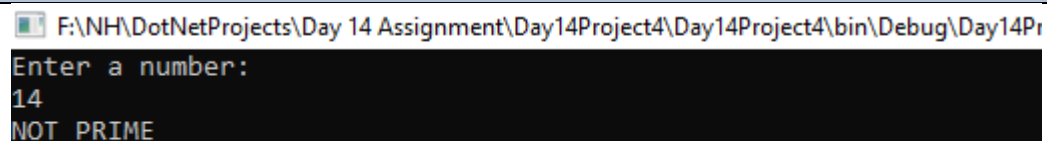**HINT : use break;**

CODE

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// ******************************************
// Author : Nanam Vaishnavi
// Purpose : Prime number using break
// ******************************************

namespace Day14Project4
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int v , i;
            Console.WriteLine("Enter a number: ");
            v = Convert.ToInt32(Console.ReadLine());
            for(i=2;i<v;i++)
            {
                if (v % i == 0)
                    break;
            }

            if (i == v)
                Console.WriteLine("PRIME");
            else
                Console.WriteLine("NOT PRIME");

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

OUTPUT

```
F:\NH\DotNetProjects\Day 14 Assignment\Day14Project4\Day14Project4\bin\Debug\Day14Pr
Enter a number:
14
NOT PRIME
```

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

CODE

```csharp
using System;
using System.Collections.Generic;
```

```csharp
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// ***************************************************************************
// Author : Nanam Vaishnavi
// Purpose : print numbers from 1 to 30 and skip the numbers divisible by 3.
//            HINT : use continue;
//***************************************************************************

namespace Day14Project5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            for (int i=1; i <= 30;i++)
            {
                if (i % 3 == 0)
                    continue;

                Console.WriteLine(i);

            }
            Console.ReadLine();

        }
    }
}
```

OUTPUT

F:\NH\DotNetProjects\Day 14 Assignment\Day14Project5\Day14Project5\bin\Debug\Day14Project5.exe

```
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

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// ********************************************************
// Author : Nanam Vaishnavi
// Purpose: Find the first number after 1000 which is divisible by 97.
//          HINT : use for loop and break
// ********************************************************
namespace Day14Project6
{
    internal class Program
    {
        static void Main(string[] args)
        {
            for(int i=1000;i<=1097; i++)
            {
                if (i % 97 == 0)
                {
                    Console.WriteLine(i);
                    break;
                }
            }
            Console.ReadLine();
        }
    } }
```

OUTPUT

F:\NH\DotNetProjects\Day 14 Assignment\Day14Project6\Day14Project6\bin\Debug\Day14Project6.exe

1067