.Net Programming

Assingment-4

Name: V. Sai Nikhil

Reg No: 16MIS0257

Question 1:

Vaccumcleanerclass.dll

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

namespace vaccumcleanerclass
{
    public class vaccum
    {
        public string vname, vmodel;
        public int vnumber;
        public vaccum()
        {

        }
        public string name
        {
            set { vname = value; }
            get { return vname; }
        }
        public string model
        {
            set { vmodel = value; }
            get { return vmodel; }
        }
        public int number
        {
            set { vnumber = value; }
            get { return vnumber; }
        }
        public void vaccumon()
        {
        }
        public void vaccumoff()
        {
        }
        public void charge()
        {
        }

    }
}
```

VaccumCleaner-Console Application:

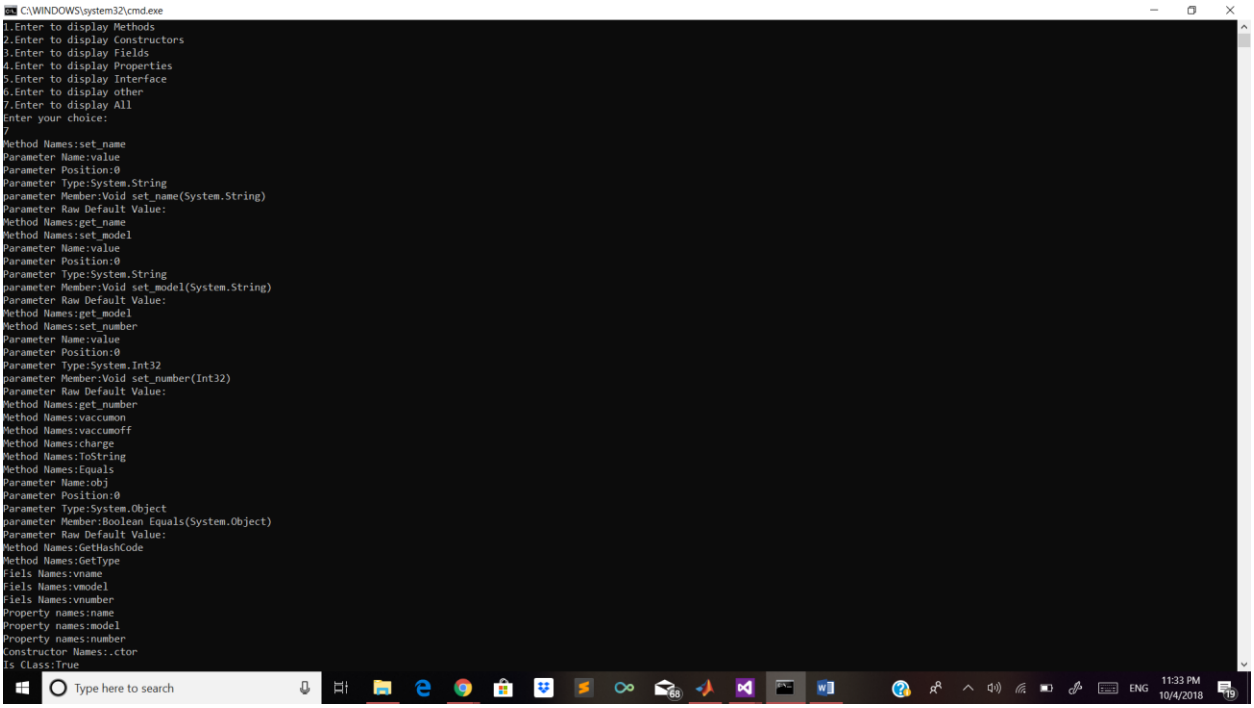
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using vaccumcleanerclass;
using Microsoft.Win32;
using System.Reflection;

public delegate void sd(object obj);

namespace vaccumcleaner
{
    public class reflect
    {
        public int mcount=0,pmcount=0,fcount=0,pcount=0,ccount=0,icount=0;
        RegistryKey rk;
        public reflect()
        {
            rk = Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\vaccumcleaner",
true);
            if (rk == null)
            {
                rk =
Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\vaccumcleaner");
            }
        }
        public void get_m(object obj)
        {
            Type t = obj.GetType();
            MethodInfo[] mi = t.GetMethods();
            foreach (MethodInfo m in mi)
            {
                Console.WriteLine("Method Names:{0}", m.Name);
                mcount++;
                ParameterInfo[] pi = m.GetParameters();
                foreach (ParameterInfo p in pi)
                {
                    Console.WriteLine("Parameter Name:{0}", p.Name);
                    pmcount++;
                    Console.WriteLine("Parameter Position:{0}", p.Position);
                    Console.WriteLine("Parameter Type:{0}", p.ParameterType);
                    Console.WriteLine("parameter Member:{0}", p.Member);
                    Console.WriteLine("Parameter Raw Default Value:{0}",
p.RawDefaultValue);
                }
                rk.SetValue("Parameter", pmcount);
            }
            rk.SetValue("Method", mcount);
        }
        public void get_f(object obj)
        {
            Type t = obj.GetType();
            FieldInfo[] fi = t.GetFields();
            foreach (FieldInfo f in fi)
```

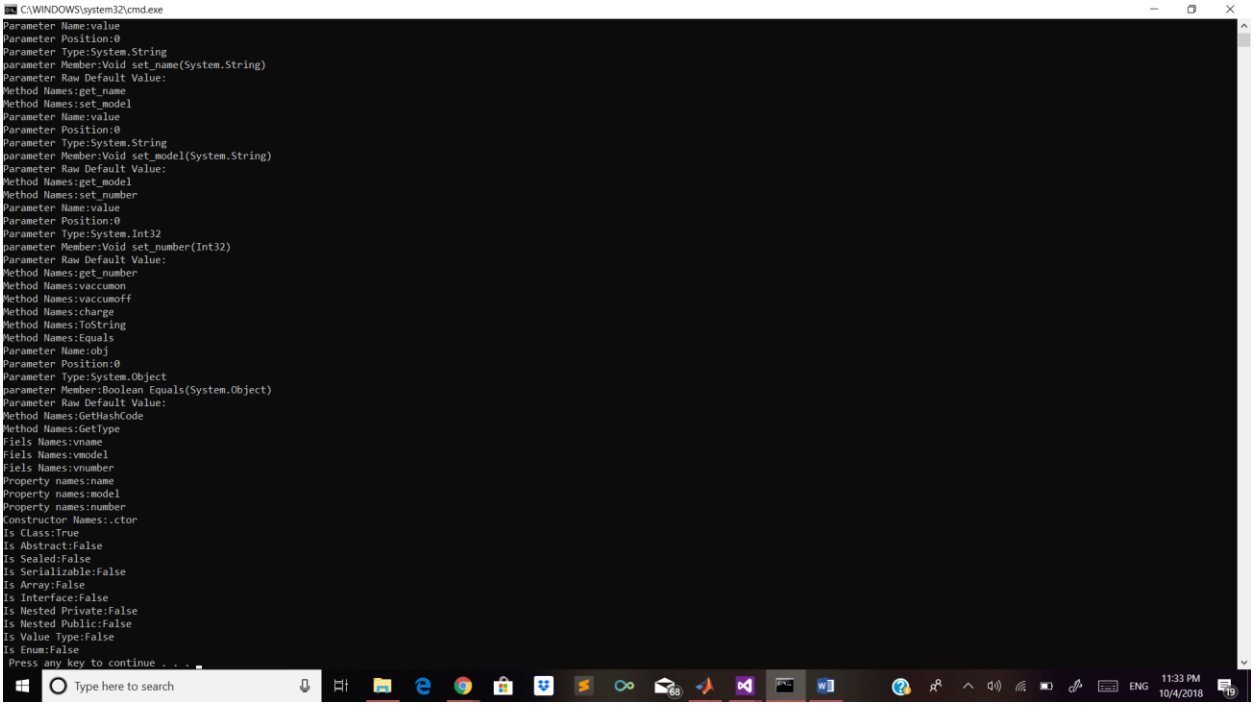
```
    {
        Console.WriteLine("Fiels Names:{0}", f.Name);
        fcount++;
    }
    rk.SetValue("Field", fcount);
}
public void get_p(object obj)
{
    Type t = obj.GetType();
    PropertyInfo[] ppi = t.GetProperties();
    foreach (PropertyInfo p in ppi)
    {
        Console.WriteLine("Property names:{0}", p.Name);
        pcount++;
    }
    rk.SetValue("Properties", pcount);
}
public void get_c(object obj)
{
    Type t = obj.GetType();
    ConstructorInfo[] ci = t.GetConstructors();
    foreach (ConstructorInfo c in ci)
    {
        Console.WriteLine("Constructor Names:{0}", c.Name);
        ccount++;
    }
    rk.SetValue("Constructor", ccount);
}
public void get_i(object obj)
{
    Type t = obj.GetType();
    Type[] ti = t.GetInterfaces();
    foreach (Type o in ti)
    {
        Console.WriteLine(o.Name);
        MethodInfo[] mi = o.GetMethods();
        foreach (MethodInfo m in mi)
        {
            Console.WriteLine("Interface Method Names:{0}", m.Name);
            icount++;
        }
        FieldInfo[] fi = o.GetFields();
        foreach (FieldInfo f in fi)
            Console.WriteLine("Interface Fiels Namws:{0}", f.Name);
        PropertyInfo[] ppi = t.GetProperties();
        foreach (PropertyInfo p in ppi)
            Console.WriteLine("Interface Property Names:{0}", p.Name);
        ConstructorInfo[] ci = t.GetConstructors();
        foreach (ConstructorInfo c in ci)
            Console.WriteLine("Interface COnstructor Names:{0}", c.Name);
    }
    rk.SetValue("Interface", icount);
}
public void get_oth(object obj)
{
    Type t = obj.GetType();
    Console.WriteLine("Is CLass:{0}", t.IsClass);
    Console.WriteLine("Is Abstract:{0}", t.IsAbstract);
```

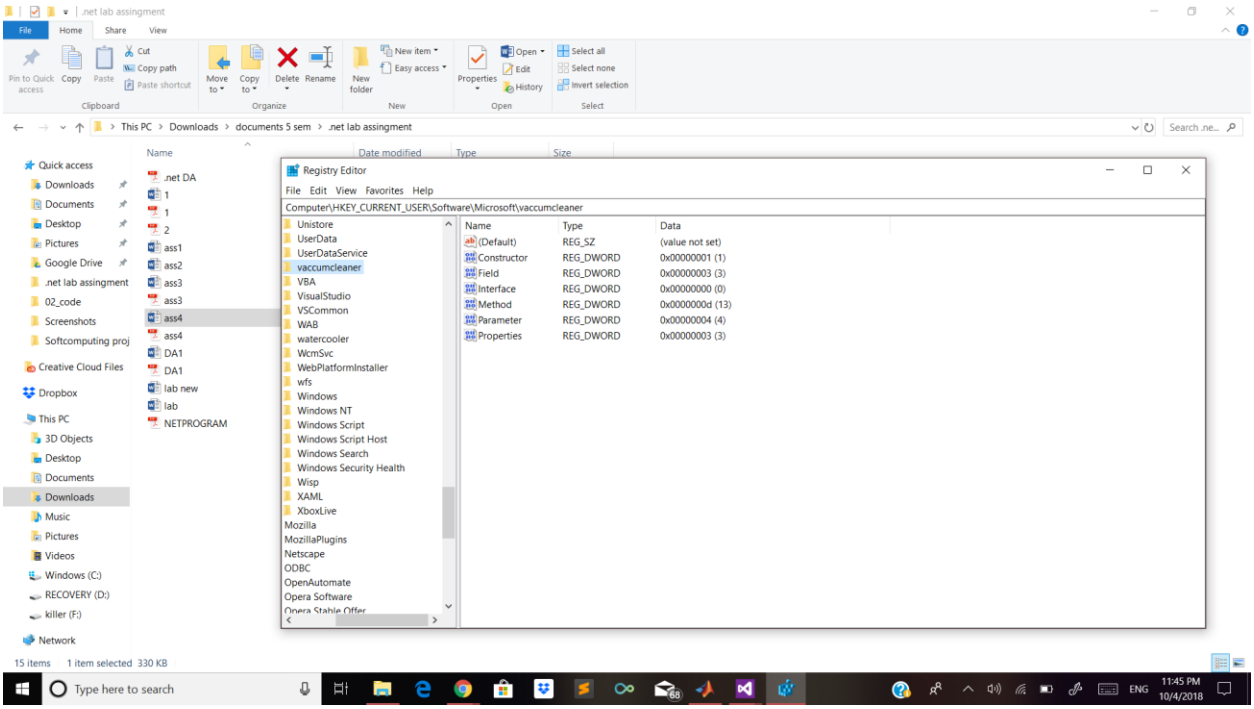
```
        Console.WriteLine("Is Sealed:{0}", t.IsSealed);
        Console.WriteLine("Is Serializable:{0}", t.IsSerializable);
        Console.WriteLine("Is Array:{0}", t.IsArray);
        Console.WriteLine("Is Interface:{0}", t.IsInterface);
        Console.WriteLine("Is Nested Private:{0}", t.IsNestedPrivate);
        Console.WriteLine("Is Nested Public:{0}", t.IsNestedPublic);
        Console.WriteLine("Is Value Type:{0}", t.IsValueType);
        Console.WriteLine("Is Enum:{0}", t.IsEnum);
    }
}

class Program
{
    static void Main(string[] args)
    {
        vaccum v = new vaccum();
        reflect g = new reflect();
        Console.WriteLine("1.Enter to display Methods");
        Console.WriteLine("2.Enter to display Constructors");
        Console.WriteLine("3.Enter to display Fields");
        Console.WriteLine("4.Enter to display Properties");
        Console.WriteLine("5.Enter to display Interface");
        Console.WriteLine("6.Enter to display other");
        Console.WriteLine("7.Enter to display All");

        Console.WriteLine("Enter your choice:");
        int ch = Convert.ToInt32(Console.ReadLine());

        switch (ch)
        {
            case 1:
                g.get_m(v);
                break;
            case 2:
                g.get_c(v);
                break;
            case 3:
                g.get_f(v);
                break;
            case 4:
                g.get_p(v);
                break;
            case 5:
                g.get_i(v);
                break;
            case 6:
                g.get_oth(v);
                break;
            case 7:
                g.get_m(v);
                g.get_f(v);
                g.get_p(v);
                g.get_c(v);
                g.get_i(v);
                g.get_oth(v);
                break;
            default:
                Console.WriteLine("Invalid choice");
```

```
                    break;
                }
                Console.ReadKey();
            }
        }
    }
```

Output:




C:\WINDOWS\system32\cmd.exe
Parameter Name:value
Parameter Position:0
Parameter Type:System.String
parameter Member:Void set_name(System.String)
Parameter Raw Default Value:
Method Names:get_name
Method Names:set_model
Parameter Name:value
Parameter Position:0
Parameter Type:System.String
parameter Member:Void set_model(System.String)
Parameter Raw Default Value:
Method Names:get_model
Method Names:set_number
Parameter Name:value
Parameter Position:0
Parameter Type:System.Int32
parameter Member:Void set_number(Int32)
Parameter Raw Default Value:
Method Names:get_number
Method Names:vaccumon
Method Names:vaccumoff
Method Names:charge
Method Names:ToString
Method Names:Equals
Parameter Name:obj
Parameter Position:0
Parameter Type:System.Object
parameter Member:Boolean Equals(System.Object)
Parameter Raw Default Value:
Method Names:GetHashCode
Method Names:GetType
Fiels Names:vname
Fiels Names:vmodel
Fiels Names:vnumber
Property names:name
Property names:model
Property names:number
Constructor Names:.ctor
Is CLass:True
Is Abstract:False
Is Sealed:False
Is Serializable:False
Is Array:False
Is Interface:False
Is Nested Private:False
Is Nested Public:False
Is Value Type:False
Is Enum:False
Press any key to continue . . .
Type here to search
ENG
11:33 PM
10/4/2018


.net lab assignment
File
Home
Share
View
This PC > Downloads > documents 5 sem > .net lab assignment
Search .ne...
Quick access
Downloads
Documents
Desktop
Pictures
Google Drive
.net lab assignment
02_code
Screenshots
Softcomputing proj
Creative Cloud Files
Dropbox
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows (C:)
RECOVERY (D:)
killer (F:)
Network
Name
.net DA
1
1
2
ass1
ass2
ass3
ass3
ass4
ass4
DA1
DA1
lab new
lab
NETPROGRAM
Registry Editor
File Edit View Favorites Help
Computer\HKEY_CURRENT_USER\Software\Microsoft\vaccumcleaner
Unistore
UserData
UserDataService
vaccumcleaner
VBA
VisualStudio
VSCommon
WAB
watercooler
WcmSvc
WebPlatformInstaller
wfs
Windows
Windows NT
Windows Script
Windows Script Host
Windows Search
Windows Security Health
Wisp
XAML
XboxLive
Mozilla
MozillaPlugins
Netscape
ODBC
OpenAutomate
Opera Software
Name Type Data
(Default) REG_SZ (value not set)
Constructor REG_DWORD 0x00000001 (1)
Field REG_DWORD 0x00000003 (3)
Interface REG_DWORD 0x00000000 (0)
Method REG_DWORD 0x0000000d (13)
Parameter REG_DWORD 0x00000004 (4)
Properties REG_DWORD 0x00000003 (3)
15 items
1 item selected 330 KB
Type here to search
ENG
11:45 PM
10/4/2018