.NET PROGRAMMING DA-1

Name: Y SRIKANTH REDDY
Reg.No:16MIS0302

VB.NET PROGRAM

```
Public Class State


    Sub Addition()

    End Sub
    Sub Multiply()
    End Sub

    Sub Subtract()



    End Sub


End Class
```

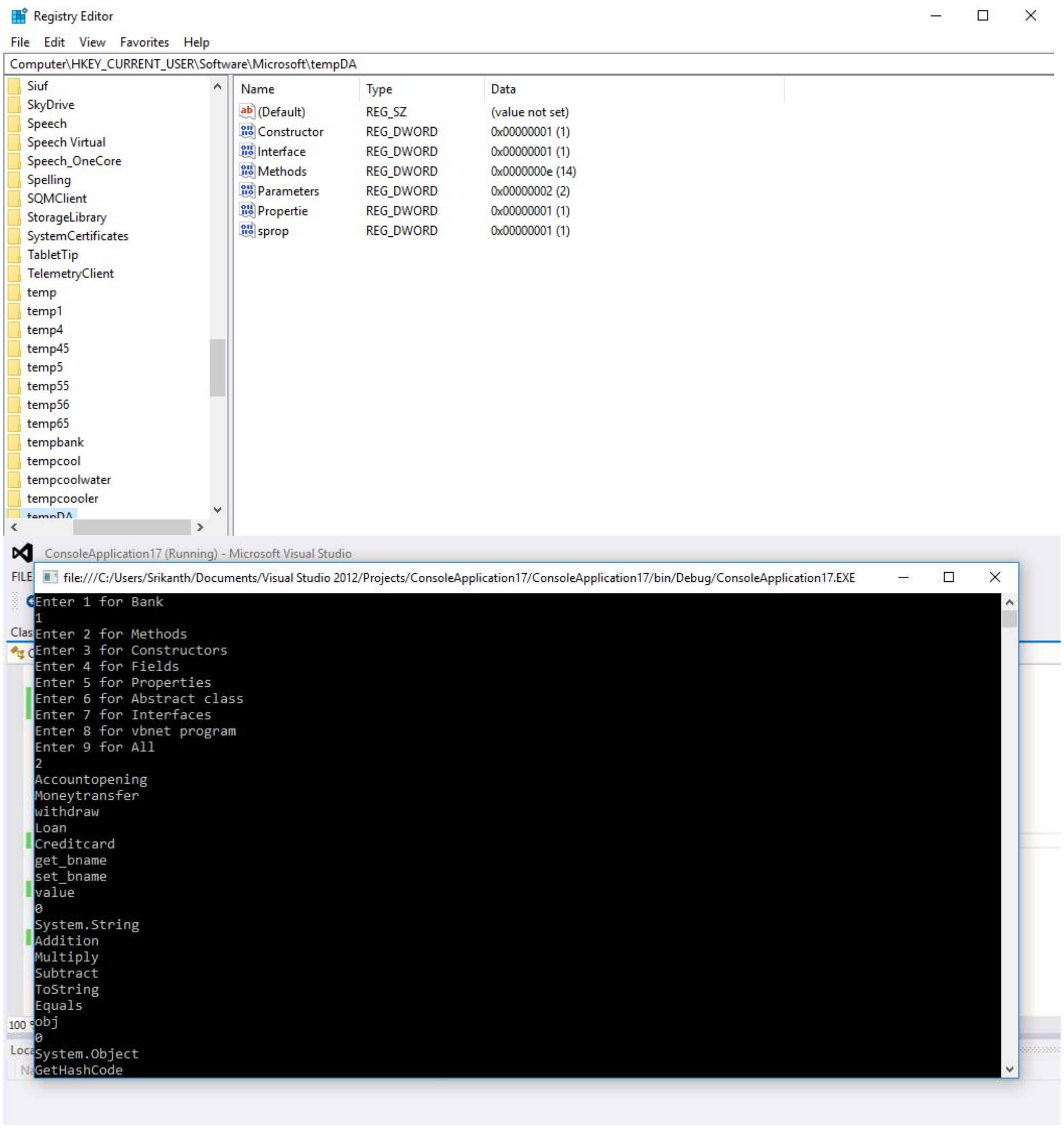
CLASS LIBRARY

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ClassLibrary15;

namespace ClassLibrary11
{
    // inheriting the Class of State   & Bank
    public class Bank:State
    {
        public string name, place;
        public double Accountno,Branchcode;
        //constructor
        public Bank()
        {

        }

        //methods
        public void Accountopening()
        {
        }
        public void Moneytransfer()
        {
        }
        public void withdraw()
        {
        }
        public void Loan()
        {
        }
        public void Creditcard()
        {
        }
        //property
        public string bname
        {
            get
            {
                return name;
            }
            set
            {
                name=value;
            }
        }



        // interface
       public class jointacount
       {
       }
       public class loanaccount
       {
       }
        public class jointaccountloanaccount
        {
        }
    }
}
```

Console Application

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ClassLibrary11;
using System.Reflection;
using Microsoft.Win32;
using ClassLibrary15;

public delegate void sd(object obj);
namespace ConsoleApplication17
{
    public class reef
    {
        RegistryKey rk;
        int mcount = 0,pmcount=0, fcount = 0, ccount = 0, pcount = 0,
procount=0,intcount=0,mmcount=0;
        public reef()
        {
            rk = Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\tempDA", true);
            if (rk == null)
            {
                rk = Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\tempDA");
            }
        }
        public void List_m(object b1)
        {
            Type t = b1.GetType();
            MethodInfo[] mi = t.GetMethods();
            foreach (MethodInfo m in mi)
            {
                Console.WriteLine(m.Name);
                mcount++;
                ParameterInfo[] pi = m.GetParameters();
                foreach (ParameterInfo p in pi)
                {
                    Console.WriteLine(p.Name);
                    Console.WriteLine(p.Position);
                    Console.WriteLine(p.ParameterType);
                    pmcount++;
                }
                rk.SetValue("Parameters", pmcount);
            }
            rk.SetValue("Methods", mcount);
            Console.WriteLine(mcount);
        }

        public void List_f(object b1)
        {
            Type t = b1.GetType();
            FieldInfo[] mi = t.GetFields();
            foreach (FieldInfo m in mi)
            {
                Console.WriteLine(m.Name);
                fcount++;

            }
            rk.SetValue("Fields", fcount);
            Console.WriteLine(fcount);

        }
        public void List_p(object b1)
        {
            Type t = b1.GetType();
            PropertyInfo[] mi = t.GetProperties();
            foreach (PropertyInfo m in mi)
            {
                Console.WriteLine(m.Name);
                pcount++;

            }
            rk.SetValue("Propertie", pcount);
            Console.WriteLine(pcount);

        }
        public void List_c(object b1)
        {
            Type t = b1.GetType();
            ConstructorInfo[] mi = t.GetConstructors();
            foreach (ConstructorInfo m in mi)
            {
                Console.WriteLine(m.Name);
                ccount++;

            }
            rk.SetValue("Constructor", ccount);
            Console.WriteLine(ccount);

        }
        public void get_interfaces(object b1)
        {
            Type t = b1.GetType();
            Type[] t1 = t.GetInterfaces();
            Type t2 = t1.GetType();
            MethodInfo[] mi = t2.GetMethods();
            foreach (MethodInfo m in mi)
            {
                Console.WriteLine(m.Name);

                ParameterInfo[] pi = m.GetParameters();
                foreach (ParameterInfo p in pi)
                {
                    Console.WriteLine(p.Name);
                    Console.WriteLine(p.ParameterType);
                    Console.WriteLine(p.Position);

                }
            }
            FieldInfo[] fi = t2.GetFields();
            foreach (FieldInfo f in fi)
            {
                Console.WriteLine(f.Name);

            }
            ConstructorInfo[] ci = t2.GetConstructors();
            foreach (ConstructorInfo c in ci)
            {
                Console.WriteLine(c.Name);

            }
            PropertyInfo[] pi1 = t2.GetProperties();
            foreach (PropertyInfo p in pi1)
            {
                Console.WriteLine(p.Name);
            }
            intcount++;
            rk.SetValue("Interface", intcount);
            Console.WriteLine(intcount);
        }
        public void get_sprop(object b1) //Reflection properties
        {
            Type t = b1.GetType();
            Console.WriteLine(t.IsAbstract);
            Console.WriteLine(t.IsArray);
            procount++;
            rk.SetValue("sprop", procount);
            Console.WriteLine(procount);

        }


    }




    class Program
    {
        static void Main(string[] args)
        {
              Assembly a = null;
            try
            {
                a = Assembly.Load("ClassLibrary11");


            }
            catch (Exception e)
            {
                Console.WriteLine(e);
            }
            reef r1 = new reef();
            State b = new State();
            //p.listalltypes(a);
            Type t = a.GetType("ClassLibrary11.Bank");
            Console.WriteLine("Enter 1 for Bank");
            int r = Convert.ToInt32(Console.ReadLine());


            switch (r)
            {
                case 1:
            Console.WriteLine("Enter 2 for Methods");
            Console.WriteLine("Enter 3 for Constructors");
            Console.WriteLine("Enter 4 for Fields");
            Console.WriteLine("Enter 5 for Properties");
            Console.WriteLine("Enter 6 for Reflection Properties");
            Console.WriteLine("Enter 7 for Interfaces");
            Console.WriteLine("Enter 8 for vbnet program");
            Console.WriteLine("Enter 9 for All");

                    int r2 = Convert.ToInt32(Console.ReadLine());
                    switch (r2)
                    {
                        case 2:
                            object c = Activator.CreateInstance(t);

                            r1.List_m(c);
```

```
                            break;
                        case 3:
                            object d = Activator.CreateInstance(t);
                            r1.List_c(d);

                            break;
                        case 4:
                            object e = Activator.CreateInstance(t);
                            r1.List_f(e);
                            break;
                        case 5:
                            object f = Activator.CreateInstance(t);
                            r1.List_p(f);
                            break;
                        case 6:
                            object g = Activator.CreateInstance(t);
                            r1.get_sprop(g);
                            break;
                        case 7:
                            object h = Activator.CreateInstance(t);
                            r1.get_interfaces(h);
                            break;
                        case 8:
                          // for methods in vb.net proogram
                            r1.List_m(b); ///   for individual of vb.net program not necessary
                                                 delete 8 case not a problem


                            break;
                        case 9:

                            object s = Activator.CreateInstance(t);
                            r1.List_m(s);
                            r1.List_c(s);
                            r1.List_p(s);
                            r1.get_sprop(s);
                            r1.get_interfaces(s);


                            break;
                        default:
                            {
                                Console.WriteLine("Please choose a valid option");
                                break;
                            }
                    }
                    break;


            }
            Console.ReadKey();


        }
    }
}
```



Registry Editor

File Edit View Favorites Help

Computer\HKEY_CURRENT_USER\Software\Microsoft\tempDA

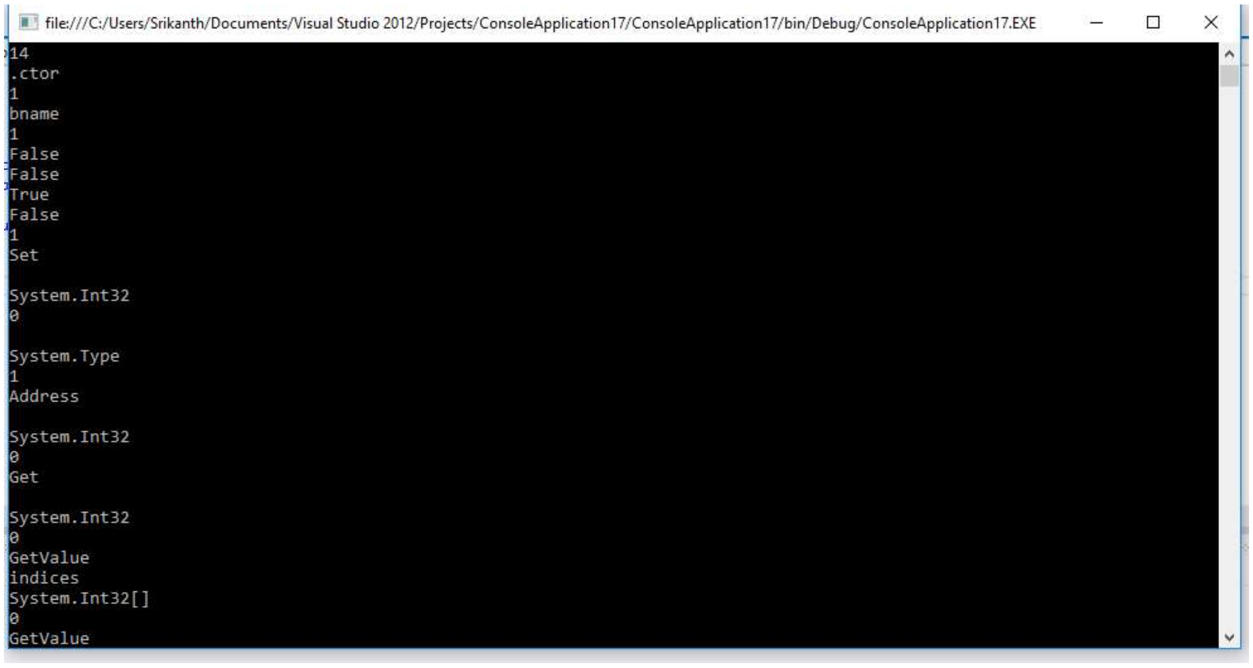
```
Siuf
SkyDrive
Speech
Speech Virtual
Speech_OneCore
Spelling
SQMClient
StorageLibrary
SystemCertificates
TabletTip
TelemetryClient
temp
temp1
temp4
temp45
temp5
temp55
temp56
temp65
tempbank
tempcool
tempcoolwater
tempcoooler
tempDA
```

Name	Type	Data
(Default)	REG_SZ	(value not set)
Constructor	REG_DWORD	0x00000001 (1)
Interface	REG_DWORD	0x00000001 (1)
Methods	REG_DWORD	0x0000000e (14)
Parameters	REG_DWORD	0x00000002 (2)
Propertie	REG_DWORD	0x00000001 (1)
sprop	REG_DWORD	0x00000001 (1)

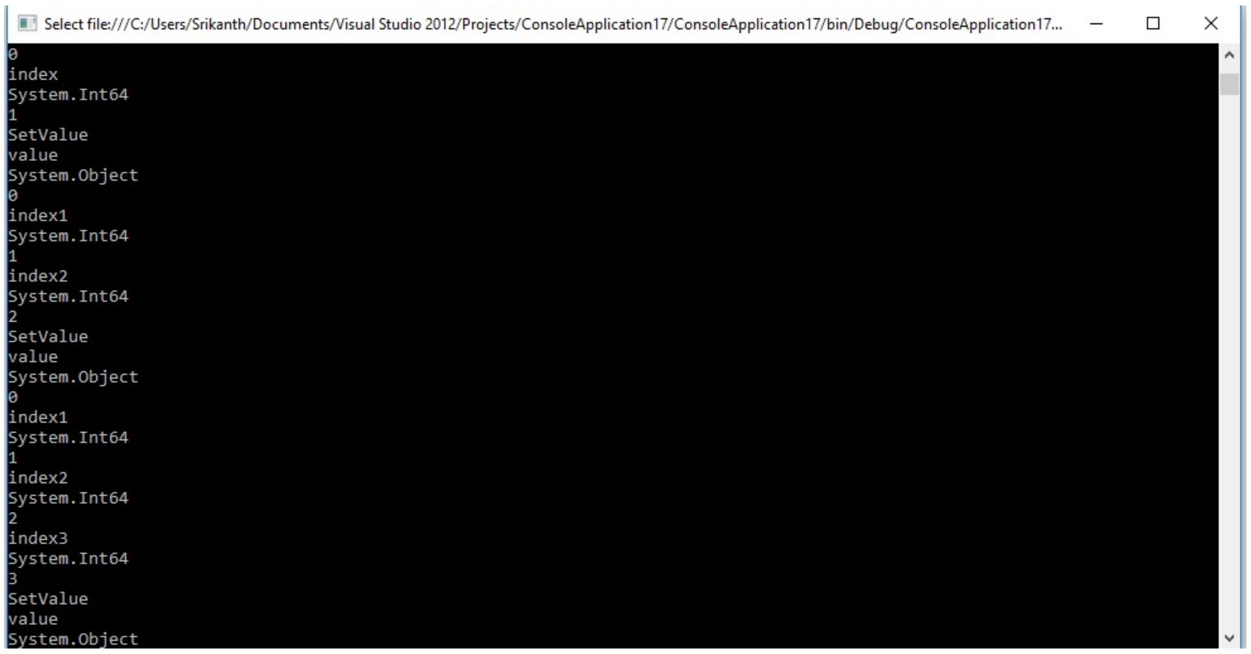
ConsoleApplication17 (Running) - Microsoft Visual Studio

file:///C:/Users/Srikanth/Documents/Visual Studio 2012/Projects/ConsoleApplication17/ConsoleApplication17/bin/Debug/ConsoleApplication17.EXE

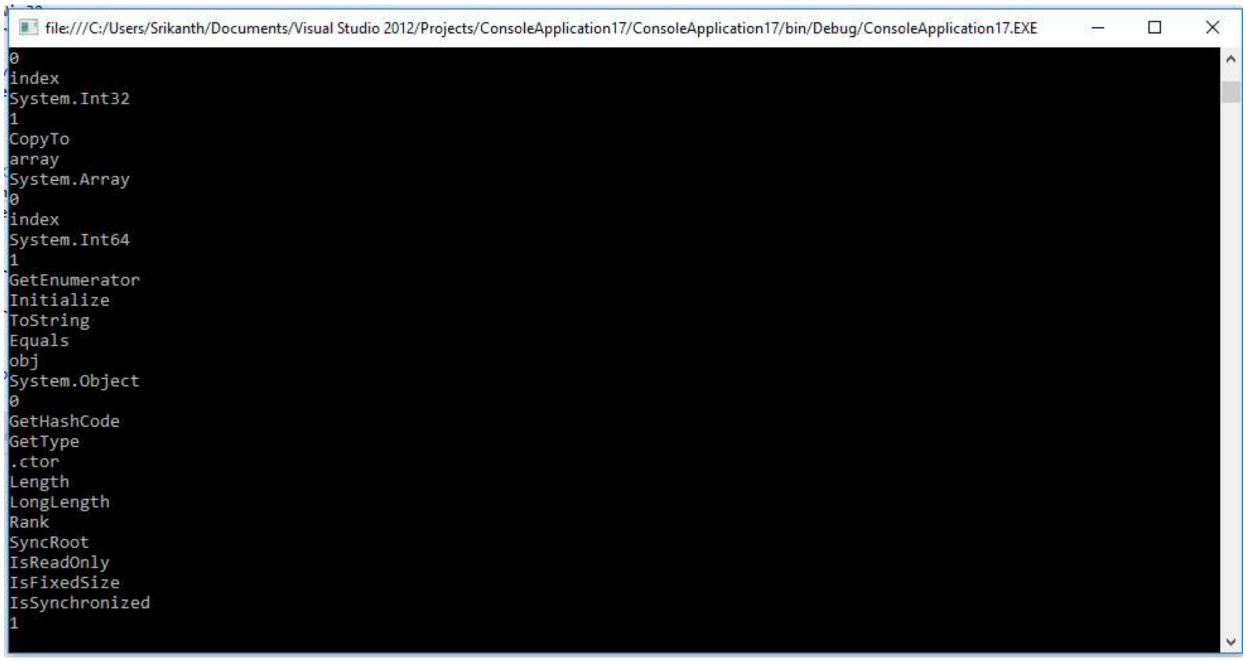
```
Enter 1 for Bank
1
Enter 2 for Methods
Enter 3 for Constructors
Enter 4 for Fields
Enter 5 for Properties
Enter 6 for Abstract class
Enter 7 for Interfaces
Enter 8 for vbnet program
Enter 9 for All
2
Accountopening
Moneytransfer
withdraw
Loan
Creditcard
get_bname
set_bname
value
0
System.String
Addition
Multiply
Subtract
ToString
Equals
obj
0
System.Object
GetHashCode
```


file:///C:/Users/Srikanth/Documents/Visual Studio 2012/Projects/ConsoleApplication17/ConsoleApplication17/bin/Debug/ConsoleApplication17.EXE

```
14
.ctor
1
bname
1
False
False
True
False
1
Set

System.Int32
0

System.Type
1
Address

System.Int32
0
Get

System.Int32
0
GetValue
indices
System.Int32[]
0
GetValue
```


Select file:///C:/Users/Srikanth/Documents/Visual Studio 2012/Projects/ConsoleApplication17/ConsoleApplication17/bin/Debug/ConsoleApplication17...

```
0
index
System.Int64
1
SetValue
value
System.Object
0
index1
System.Int64
1
index2
System.Int64
2
SetValue
value
System.Object
0
index1
System.Int64
1
index2
System.Int64
2
index3
System.Int64
3
SetValue
value
System.Object
```



file:///C:/Users/Srikanth/Documents/Visual Studio 2012/Projects/ConsoleApplication17/ConsoleApplication17/bin/Debug/ConsoleApplication17.EXE

```
0
index
System.Int32
1
CopyTo
array
System.Array
0
index
System.Int64
1
GetEnumerator
Initialize
ToString
Equals
obj
System.Object
0
GetHashCode
GetType
.ctor
Length
LongLength
Rank
SyncRoot
IsReadOnly
IsFixedSize
IsSynchronized
1
```