

Programming in C# Lab 2

Objectives:

At the end of this session, you will be able to understand:

- ❖ *Statements & Operators*
- ❖ *Programming Construct*
- ❖ *Array*

Part I: Workshop – 15 minutes

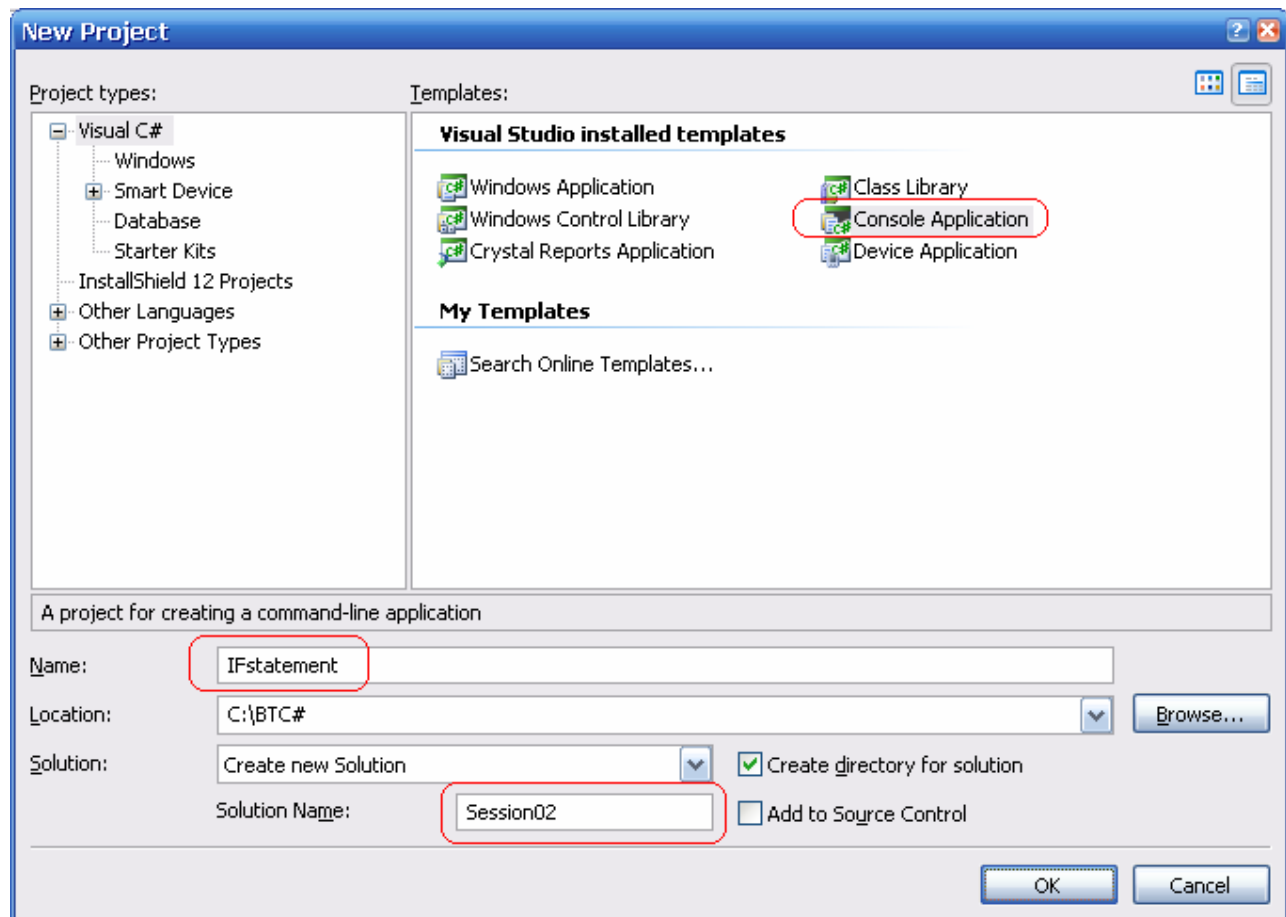
Students open workshop in CD ROM, then View, Run, Think about it.

Part II: Step by step – 45 minutes

Exercise 1: IF statement

Step 1: Open Visual Studio 2005

Step 2: Select the menu File->New->Project to create console based project named 'IFstatement' and Solution named Session02 as following



Step 3: Rename the class file 'program.cs' to 'IFstatement.cs'

Step 4: Replace code in 'IFStatement.cs' with given code

```
using System;
class IFDemo
{
    public static void Main(String[] args)
    {
        double a, b, c, delta, x1, x2;
        //input a,b,c
        Console.WriteLine("a=");
        //a phai !=0
        do
        {
            a = Convert.ToInt32(Console.ReadLine());
        }
        while (a == 0);
        Console.WriteLine("b=");
        b = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("c=");
        c = Convert.ToInt32(Console.ReadLine());

        delta = b * b - 4 * a * c;

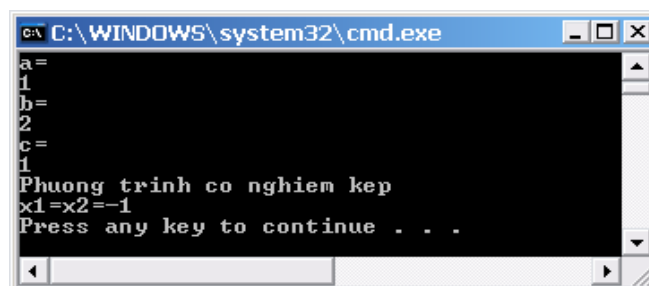
        if (delta < 0)
            Console.WriteLine("Phuong trinh vo nghiem");
        else if (delta == 0)
        {
            Console.WriteLine("Phuong trinh co nghiem kep");
            Console.WriteLine("x1=x2={0}", -b / (2 * a));
        }
        else
        {
            Console.WriteLine("Phuong trinh co 2 nghiem");
            x1 = (-b + Math.Sqrt(delta)) / (2 * a);
            Console.WriteLine("x1={0}", x1);
            x2 = (-b - Math.Sqrt(delta)) / (2 * a);
            Console.WriteLine("x2={0}", x2);
        }
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build IFstatement option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of the program as following



```
C:\WINDOWS\system32\cmd.exe
a=
1
b=
2
c=
1
Phuong trinh co nghiem kep
x1=x2=-1
Press any key to continue . . .
```

Exercise 2: Switch statement

Step 1: Add a console based project 'SwitchStatement' to the solution

Step 2: Right click on project SwitchStatement -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'SwitchStatement.cs'

Step 4: Replace the code in 'SwitchStatement.cs' with the given code

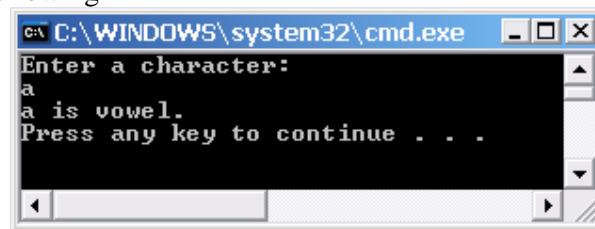
```
using System;
class SwitchDemo
{
    public static void Main(String[] args)
    {
        char ch;
        Console.WriteLine("Enter a character:");
        ch = (char)Console.Read();
        switch (ch)
        {
            case 'a':
            case 'o':
            case 'e':
            case 'u':
            case 'i': Console.WriteLine("{0} is vowel.", ch);
                     break;
            default: Console.WriteLine("{0} is consonant.", ch);
                     break;
        }
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'SwitchStatement' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



Exercise 3: for statement

Step 1: Add a console based project 'ForStatement' to the solution

Step 2: Right click on project ForStatement -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'ForStatement.cs'

Step 4: Replace the code in 'ForStatement.cs' with the given code

```
using System;
```

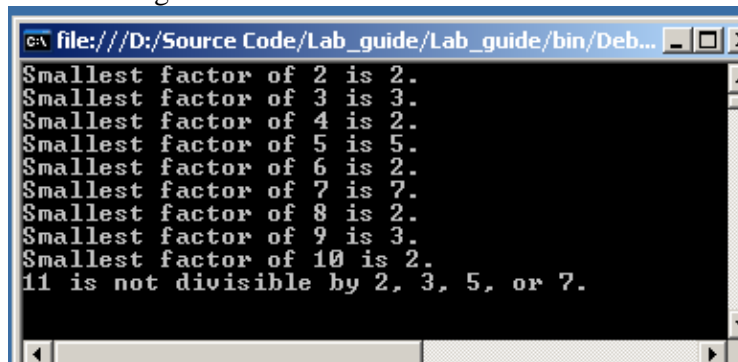
```
class ForDemo
{
    static void Main(string[] args)
    {
        int num;
        for (num = 2; num < 12; num++)
        {
            if ((num % 2) == 0)
                Console.WriteLine("Smallest factor of " + num + " is 2.");
            else if ((num % 3) == 0)
                Console.WriteLine("Smallest factor of " + num + " is 3.");
            else if ((num % 5) == 0)
                Console.WriteLine("Smallest factor of " + num + " is 5.");
            else if ((num % 7) == 0)
                Console.WriteLine("Smallest factor of " + num + " is 7.");
            else
                Console.WriteLine(num + " is not divisible by 2, 3, 5, or 7.");
        }
        Console.ReadLine();
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ForStatement' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



Exercise 4: foreach statement

Step 1: Add a console based project 'ForEach' to the solution

Step 2: Right click on project ForEach -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'ForEach.cs'

Step 4: Replace the code in 'ForEach.cs' with the given code

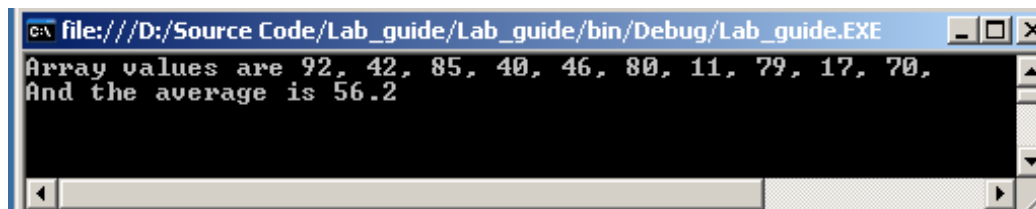
```
using System;
class ForEach
{
    static void Main(string[] args)
    {
        DateTime now = DateTime.Now;
        Random rand = new Random((int)now.Millisecond);
        int[] Arr = new int[10];
        for (int x = 0; x < Arr.Length; ++x)
        {
            Arr[x] = rand.Next() % 100;
        }
        int Total = 0;
        Console.Write("Array values are ");
        foreach (int val in Arr)
        {
            Total += val;
            Console.Write(val + ", ");
        }
        Console.WriteLine("\nAnd the average is {0,0:F1}",
                          (double)Total / (double)Arr.Length);
        Console.ReadLine();
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ForEach' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



Exercise 5: Array

Step 1: Add a console based project 'ArrayDemo1' to the solution

Step 2: Right click on project ArrayDemo1 -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'ArrayDemo1.cs'

Step 4: Replace the code in 'ArrayDemo1.cs' with the given code

```
using System;
class ArrayDemo1
{
    static void Main(string[] args)
    {
        int i;
        int[] nums1 = new int[10];
        int[] nums2 = new int[10];
```

```
for (i = 0; i < 10; i++) nums1[i] = i;

for (i = 0; i < 10; i++) nums2[i] = -i;

Console.Write("Here is nums1: ");
for (i = 0; i < 10; i++)
    Console.Write(nums1[i] + " ");
Console.WriteLine();

Console.Write("Here is nums2: ");
for (i = 0; i < 10; i++)
    Console.Write(nums2[i] + " ");
Console.WriteLine();
nums2 = nums1; // now nums2 refers to nums1
Console.Write("Here is nums2 after assignment: ");
for (i = 0; i < 10; i++)
    Console.Write(nums2[i] + " ");
Console.WriteLine();
// now operate on nums1 array through nums2
nums2[3] = 99;
Console.Write("Here is nums1 after change through nums2: ");
for (i = 0; i < 10; i++)
    Console.Write(nums1[i] + " ");
Console.WriteLine();

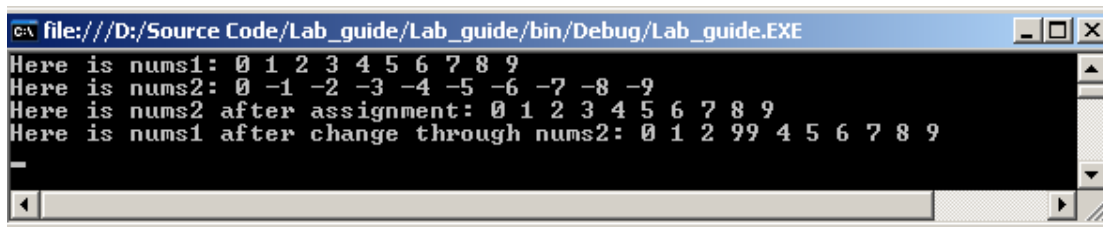
Console.ReadLine();
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ArrayDemo1' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



Exercise 5: reverse this array

Step 1: Add a console based project 'ReverseArray' to the solution

Step 2: Right click on project ReverseArray -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'ReverseArray.cs'

Step 4: Replace the code in 'ReverseArray.cs' with the given code

```
using System;
class ReverseArray
```

```
{
    static void Main(string[] args)
    {
        int i, j;
        int[] nums1 = new int[10];
        int[] nums2 = new int[10];

        for (i = 0; i < nums1.Length; i++) nums1[i] = i;

        Console.Write("Original contents: ");
        for (i = 0; i < nums2.Length; i++)
            Console.Write(nums1[i] + " ");

        Console.WriteLine();

        // reverse copy nums1 to nums2
        if (nums2.Length >= nums1.Length) // make sure nums2 is long
            enough
            for (i = 0, j = nums1.Length - 1; i < nums1.Length; i++, j--)
                nums2[j] = nums1[i];

        Console.Write("Reversed contents: ");
        for (i = 0; i < nums2.Length; i++)
            Console.Write(nums2[i] + " ");

        Console.WriteLine();

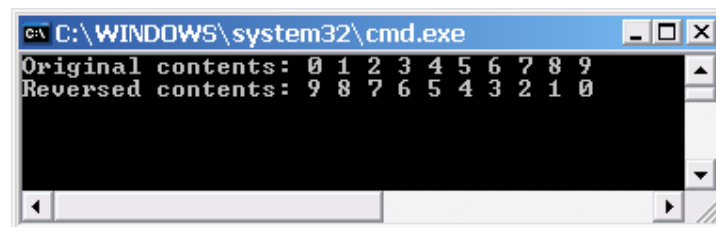
        Console.ReadLine();
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ReverseArray' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



Exercise 5: Using multi-dimensional array

Step 1: Add a console based project 'MulDimArray' to the solution

Step 2: Right click on project MulDimArray -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'MulDimArray y.cs'

Step 4: Replace the code in 'MulDimArray.cs' with the given code

```
using System;
class MulDimArray
{
```

```
static void Main(string[] args)
{
    const int rows = 4;
    const int columns = 3;

    // declare a 4x3 integer array
    int[,] rectangularArray = new int[rows, columns];

    // populate the array
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            rectangularArray[i, j] = i + j;
        }
    }

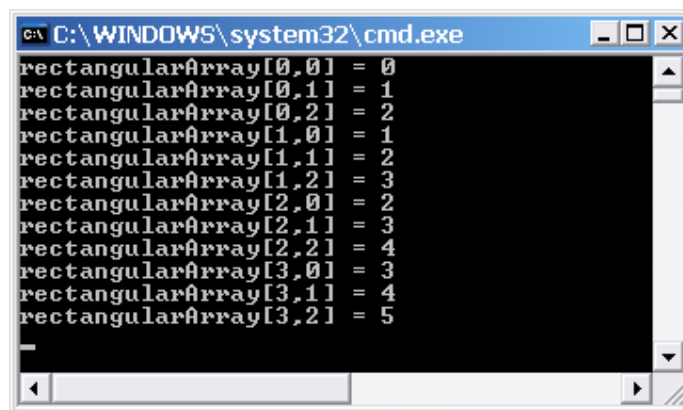
    // report the contents of the array
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < columns; j++)
        {
            Console.WriteLine("rectangularArray[{0},{1}] = {2}",
                               i, j, rectangularArray[i, j]);
        }
    }
    Console.ReadLine();
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'MulDimArray' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



```
C:\WINDOWS\system32\cmd.exe
rectangularArray[0,0] = 0
rectangularArray[0,1] = 1
rectangularArray[0,2] = 2
rectangularArray[1,0] = 1
rectangularArray[1,1] = 2
rectangularArray[1,2] = 3
rectangularArray[2,0] = 2
rectangularArray[2,1] = 3
rectangularArray[2,2] = 4
rectangularArray[3,0] = 3
rectangularArray[3,1] = 4
rectangularArray[3,2] = 5
```

Exercise 5: Array Class

Step 1: Add a console based project 'ArrayClass' to the solution

Step 2: Right click on project ArrayClass -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'ArrayClass.cs'

Step 4: Replace the code in 'ArrayClass.cs' with the given code

```
using System;
class ArrayClass
{
    static void Main(string[] args)
    {
        int[] Arr = new int[12] { 29, 82, 42, 46, 54, 65, 50, 42, 5, 94,
19, 34 };
        Console.WriteLine("The first occurrence of 42 is at index "
            + Array.IndexOf(Arr, 42));
        Console.WriteLine("The last occurrence of 42 is at index "
            + Array.LastIndexOf(Arr, 42));

        int x = 0;
        while ((x = Array.IndexOf(Arr, 42, x)) >= 0)
        {
            Console.WriteLine("42 found at index " + x);
            ++x;
        }
        x = Arr.Length - 1;
        while ((x = Array.LastIndexOf(Arr, 42, x)) >= 0)
        {
            Console.WriteLine("42 found at index " + x);
            --x;
        }

        Console.WriteLine("Array that befor sorted");
        for (int i = 0; i < Arr.Length; i++)
        {
            Console.WriteLine("{0} :      {1}", i + 1, Arr[i]);
        }
        Array.Sort(Arr);
        Console.WriteLine("Array that after sorted");
        for (int i = 0; i < Arr.Length; i++)
        {
            Console.WriteLine("{0} :      {1}", i + 1, Arr[i]);
        }
        Array.Reverse(Arr);
        Console.WriteLine("Array that after reserse");
        for (int i = 0; i < Arr.Length; i++)
        {
            Console.WriteLine("{0} :      {1}", i + 1, Arr[i]);
        }
        Console.ReadLine();
    }
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ArrayClass' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following

```

C:\WINDOWS\system32\cmd.exe
2 :      19
3 :      29
4 :      34
5 :      42
6 :      42
7 :      46
8 :      50
9 :      54
10 :     65
11 :     82
12 :     94
Array that after reserse
1 :      94
2 :      82
3 :      65
4 :      54
5 :      50
6 :      46
7 :      42
8 :      42
9 :      34
10 :     29
11 :     19
12 :      5

```

Exercise 5: Array of Object

Step 1: Add a console based project 'ObjectArray' to the solution

Step 2: Right click on project ObjectArray -> set as Startup project

Step 3: Rename the class file 'Program.cs' to 'ObjectArray.cs'

```

using System;
class Employee
{
    private int empID;

    // constructor
    public Employee(int empID)
    {
        this.empID = empID;
    }
    public override string ToString()
    {
        return empID.ToString();
    }
}
class ObjectArray
{
    public void Run()
    {
        int[] intArray;
        Employee[] empArray;
        intArray = new int[5];
        empArray = new Employee[3];

        // populate the array
        for (int i = 0; i < empArray.Length; i++)
        {
            empArray[i] = new Employee(i + 5);
        }
    }
}

```

```
Console.WriteLine("The int array...");
for (int i = 0; i < intArray.Length; i++)
{
    Console.WriteLine(intArray[i].ToString());
}

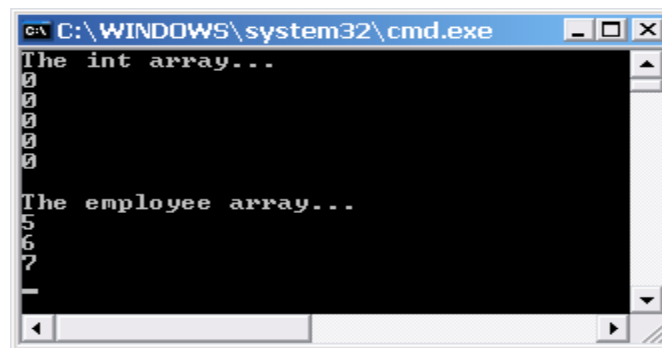
Console.WriteLine("\nThe employee array...");
for (int i = 0; i < empArray.Length; i++)
{
    Console.WriteLine(empArray[i].ToString());
}
}
static void Main(string[] args)
{
    ObjectArray arr = new ObjectArray();
    arr.Run();
    Console.ReadLine();
}
}
```

Step 5: Select menu File -> Save to save the file

Step 6: Select Build -> Build 'ObjectArray' option to build the project

Step 7: Select Debug -> Start without Debugging to execute the program

The output of program as following



Part III: Do it yourself – 60 minutes

Exercise 1: SIN Validator

Background

A Canadian SIN has nine digits. The right-most digit is a check digit that enables validation. For the whole number to be a valid SIN, a weighted sum of the other digits plus this check digit must be divisible by 10.

To obtain the weighted sum, take the digit to the left of the check digit and then every second digit leftwards. Add each of these digits to itself. Then, add each digit of each sum to form the weighted sum of the even positioned digits. Add each of the remaining SIN digits (except the check digit) to form the sum of the odd positioned digits. Add the two sums and subtract the next highest number ending in zero

from their total. If this number is the check digit, the whole number is a valid SIN; otherwise, the whole number is not a valid SIN.

Consider the following example:

```
SIN 193 456 787
      |
check digit is 7
add first set of alternates to themselves
  9 4 6 8
  9 4 6 8
  18 8 12 16
add the digits of each sum 1+8+8+1+2+1+6 = 27
add the other alternates 1+3+5+7 = 16
total = 43
Next highest integer multiple of 10 = 50
Difference = 7
Matches the check digit, therefore this SIN is valid
```

Specifications

Design a program that validates a Canadian Social Insurance Number (SIN). Your program keeps accepting a whole number and determining if that whole number is a valid SIN. Your program terminates when the user enters 0 as the whole number. The output from your program looks something like:

```
SIN Validator
=====
SIN (0 to quit): 193456787
This is a valid SIN.
SIN (0 to quit): 193456788
This is not a valid SIN.
SIN (0 to quit): 0
Have a Nice Day!
```

Part IV: Homework

Exercise 1: Do assignment of module 4 in CD ROM

Exercise 2: Do assignment of module 5 in CD ROM

References

- 1) CD ROM C# Programming, Aptech Education
- 2) <http://www.java2s.com/Tutorial/CSharp/CatalogCSharp.htm>
- 3) MSDN Document