# Arrays

ITRW211 13 Feb 2017

## 1. Input Form

Create an application that input test scores and store them in array. Display the inputted test scores in a message box. Determine the average of the scores, the highest and the lowest values and display them in textboxes.

Example 1 (Scenario 1) The number of test scores and the test scores are known beforehand.

Start Visual Studio.NET 2015 and create a new C# Windows Application project by selecting

**File | New | Project.**

1. Create an windows application(Project) in Visual C# and called it ArrayExampleApp1
2. Rename Form1.cs to MainForm.cs.
3. Determine the highest and the lowest by sorting the array in ascending order (See Example 7-6 Textbook page 408). The first element in the array will be the lowest and the last element, the highest.

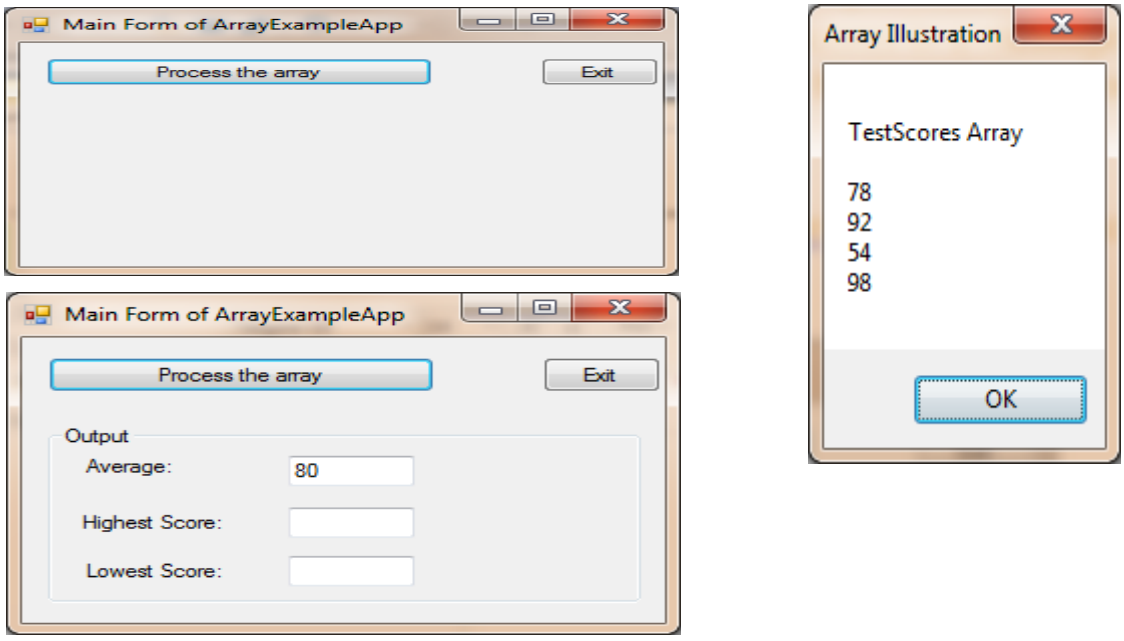
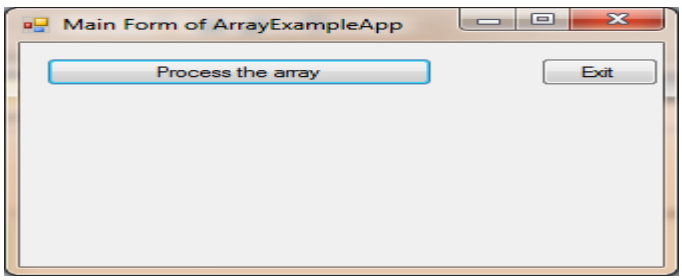


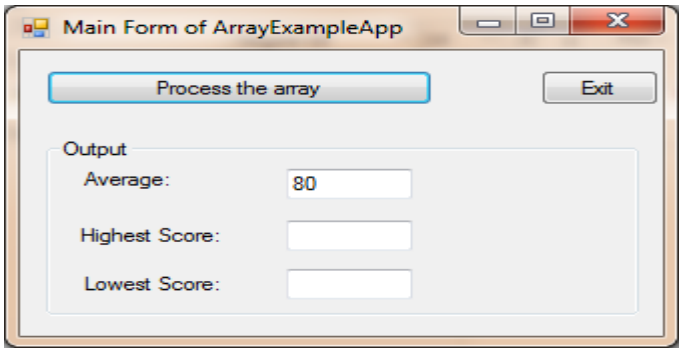

**Figure 1.1: ArrayExampleApp1**

```
private void btnProcess_Click(object sender, EventArgs e)
        {
            groupBox1.Visible = true;

            //Declare variables
            string outputMsg = "";
            int score;
            int sumScore = 0;
            int[] arrScore = { 78, 92, 54, 98 }; //creates an array and initialize
                                                                        values

            //Display contents of Array
            outputMsg += "TestScores Array \n\n";

            foreach (int val in arrScore)
                outputMsg += val + "\n";

            MessageBox.Show(outputMsg, "Array Illustration");

            for (int i = 0; i < 4; i++)
            {
                sumScore = sumScore + arrScore[i];

            }

            //Display average score
            txtAverage.Text = Convert.ToString(sumScore / 4);

        }
```

## Example 2 (Scenario 2) The number of test scores is known beforehand and the test scores are unknown.

1. Create an windows application(Project) in Visual C# and called it ArrayExampleApp1
2. Rename Form1.cs to MainForm.cs .
3. Add another form to the project and rename it InputForm.cs. We are going to use this form to get input from the user.



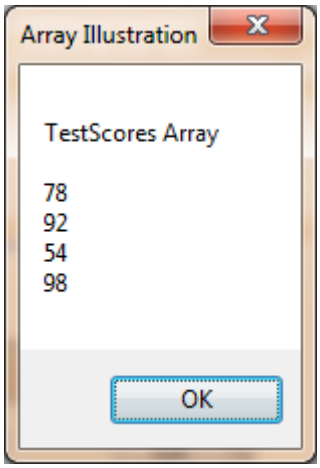




**Figure 2.1 ArrayExampleApp2**

//Main

```
        private void btnProcess_Click(object sender, EventArgs e)
        {
            groupBox1.Visible = true;

            //Declare variables
            string outputMsg = "";
            int score;
            int sumScore = 0;
            int[] arrScore = new int[4]; //creates an array of lenght 4

            //Input test scores
            InputForm Input = new InputForm(); //create the input form

            for (int i = 0; i < 4; i++)
            {
                Input.ShowDialog();  //display the input form
                score = Input.s;  //access a variable in InputForm
                arrScore[i] = score;
                sumScore = sumScore + arrScore[i]; //accumulates the scores
            }

            //Display contents of Array
            outputMsg += "TestScores Array \n\n";
            foreach (int val in arrScore)
                outputMsg += val + "\n";
            MessageBox.Show(outputMsg, "Array Illustration");


            //Display average score
            txtAverage.Text = Convert.ToString(sumScore / 4);

        }
```

//Input

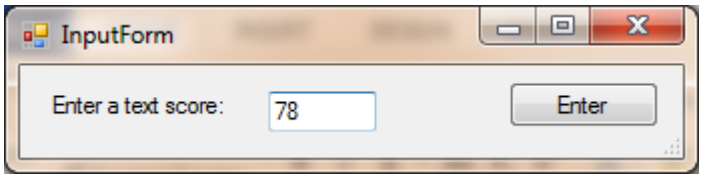



**Figure 2.2 ArrayExampleApp2**

```
using System.Windows.Forms;

namespace ArrayExampleApp
{
    public partial class frmInput : Form
    {
        public int s { get; set; }

        public frmInput()
        {
            InitializeComponent();
```

```
        }

        private void btnEnter_Click(object sender, EventArgs e)
        {
            txtScore.Focus();
            s = Convert.ToInt32(txtScore.Text);
            txtScore.Clear();
            this.Close();

        }
    }
}
```

## Exercise 1 (Scenario 3) The number of test scores and the test scores are unknown.

Modify the examples given above by using the strategies demonstrated by example 7-7.

See Example 7-7 Textbook Page 411.

## Example 3 Arrays in classes.

Arrays can be used as fields or instance variables in classes

```
/* Player.cs     Author:  Doyle
* Creates class with characteristics about one player. Includes name and ID fields,
*plus points scored. Any number of games can be used to instantiate an object of this
* *class. Average calculated based on the number of points associated with   one
*player.
 */

using System;

namespace PlayerApp
{
    public class Player
    {
        private string lname;
        private string fname;
        private string id;
        private int[ ] pointsScored;
        private int numberOfGames;

        // Default Constructor
        public Player()
              {
              }

        // Constructor accepts any size
        // pointsScored array.
        public Player (string ln, string fn,
                   string iden,  int [ ] s, int numGames)
        {
            numberOfGames = numGames;
            FillPointsScoredArray(s); //call a method that fills the array
            lname = ln;
            fname = fn;
            id = iden;
```

```
}


public string FName
{
    get
    {
        return fname;
    }
    set
    {
        fname = value;
    }
}

public string LName
{
    get
    {
        return lname;
    }
    set
    {
        lname = value;
    }
}

public string ID
{
    get
    {
        return id;
    }
    set
    {
        id = value;
    }
}

public int NumberOfGames
{
    get
    {
        return numberOfGames;
    }
    set
    {
        numberOfGames = value;
    }
}

public int[] PointsScored
{
    get
    {
        return pointsScored;
    }
    set
    {
        pointsScored = value;
    }
}
```

```
        public void FillPointsScoredArray(int [ ] s)
        {
            pointsScored = new int [numberOfGames];
            for (int i = 0; i < pointsScored.Length; i++)
                pointsScored[i] = s[i];
        }

        public double ComputeAverage( )
        {
            double total = 0;
            double avg;

            foreach(int s in pointsScored)
                    total += s;
            if (pointsScored.Length > 0)
                avg = total / pointsScored.Length;
            else
                avg = 0;
            return avg;
        }

        public override string ToString()
        {
            return "Player Name: " + fname + " " + lname +
                   "\nPlayer ID: " + id +
                   "\nNumber of Games: " + numberOfGames +
                          "\nAverage Points Scored per Game: " +
      ComputeAverage().ToString("F2");
        }
       }
}
```

Main Application that uses the class

Convert the console application given in the text book by creating an input form and a main form.

See Example 7-15 Textbook Page 430.