CO1404: Introduction to Programming

Lab 6 : Arrays and Coursework Preparation

Lab Summary

The lab worksheets are designed to help you write your first programs. Students who have experience programming should find it straightforward at first. However, there will be advanced questions later on. There's a lot of reading in this worksheet as we introduce the tools required. Later worksheets will tend to be shorter, with more work for you to do!

Remember:

- If you have any questions or require assistance, please get your tutors attention. We are all here to help!
- Refer to previous weeks materials for guidance.

Objectives

This Lab aims to achieve the following objective(s) below:

- **PO.1 :** Implementing Arrays
- **PO.2 :** Prepare for Coursework!

Lab Activity

This lab will explore how we create and interact with arrays. It consists again of just a single stage. As usual this is required to be completed by the end of this week. As always refer to the 4 **L**s: Lectures, Lecture Notes, Lab sheets and Lab Projects.

Arrays

Create a new Visual Studio C# Console project called `Arrays`. Write your answers for all these exercises in the same project. Separate each answer with:

```
Console.ReadLine();
Console.Clear();
```

Simple Arrays

1. Create an array of strings called weekdays containing the days of the week `"Sunday", "Monday", etc.)`. Read the lecture notes on how to do this.

2. Display each of the 7 array elements using 7 `WriteLine` statements in the style shown in the lecture.

3. Then use a for loop to display the 7 elements, following the style shown in the lecture - This shows how well suited for loops are for arrays.

Strings as Arrays

This exercises uses the `weekdays` array again. This time we want to display only the days that begin with `S`' or `T`. This is easy once you realise that `strings` can be used like arrays of `characters`:

- If we have a `string` called `dayString` that contains the word `Monday`
- Then `dayString[0]` is `M`, `dayString[1]` is `o`, `dayString[2]` is `n`, etc.
- Also `dayString.Length` is `6`

1. As you should be using the same project as the first exercise, you should still be able to use your earlier `weekdays` array. However, if you created a new project for some reason then you will need to declare the `weekdays` array again.

2. Write another for loop that goes through the 7 weekdays array elements, but instead of displaying them inside the loop write this line inside the loop (but use the correct variable names for your project):

   ```
   string dayString = weekdays[index];
   ```

   > **Note:** This simply copies the weekday into a local string so it is easier to work with.

3. After that line, still inside your loop, use an if statement to test whether the first character of the string is `S` **or** `T`.

   > **Note:** These are characters - we have not used characters much - remember that you use single quotation marks **'** for characters.

4. If the first character is `S` or `T`, then display that day of the week. The correct output should be (assuming you started your week days on Sunday):

   ```
   Sunday
   Tuesday
   Thursday
   Saturday
   ```

Using strings as arrays allows you to do more complex string testing. However, ***you cannot change the characters of strings using the array style**.* You must use other C# features to do that (for example there is a `dayString.Replace()` method)

Array from User Input

1. Ask the user for an array size and let them input an integer. Don't worry about validating the input, assume it will be sensible.

2. Declare an array of integers called `userArray` of the size the user asked for. You need to specify the size of the array, but do not attempt to specify the actual values it contains refer to the lecture on how to do this.

3. Use a for loop to step through and input each array element from the user. That means inside the loop you would need a line like this, but make sure you use the correct variable name for your code:

   ```
   userArray[index] = int.Parse(Console.ReadLine());
   ```

4. Write a second for loop to display the contents of this array ***in reverse order***

   > **Note:** this is similar to the earlier exercise to display an array, but this time you need a loop that decreases. **Be very careful** about choosing the start value and ending condition for this loop. The wrong choice will result in a crash or won't show all the numbers. Read the lecture notes "Out of Range Indexes" and look carefully at the range of valid indexes used. Ask your tutor if you get problems here.

Advanced Challenge:

There is an easy way to sort a simple array like this into increasing order. However, you will need to research available C# methods. Can you find it? If so then display the sorted array.

Time for Review and Coursework Preparation

Use the remainder of this lab session to go back and review previous labs that you were unsure about or didn’t complete. Try the advanced sections that you didn’t do first time round, or attempt to reach Stage 2 to push your skill further. Also use this time to ask questions of your tutor on topics you feel weak or need extra help on. Alternatively, work on your Coursework!