

The Arrays

Contents of Lecture:

- ❖ Introduction
- ❖ Reasons for Using Arrays
- ❖ Examples
- ❖ **Homework number (7) delivered next week**

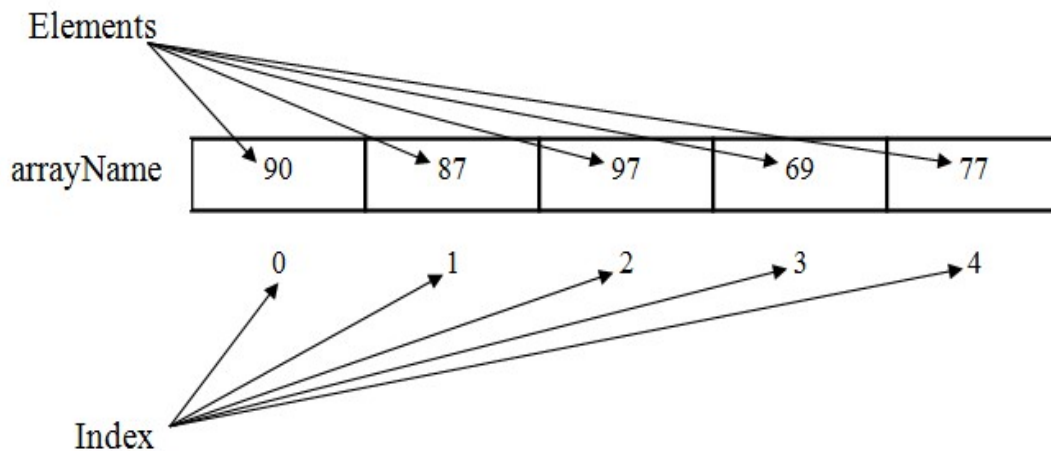
Introduction

- ❖ **Arrays** are specialized data structures which allow us to build a collection of variables are called **elements** and they are all referred to by the same array name.
- ❖ What differentiates the elements in the array is an index. The index acts as an address to uniquely identify each element in the array.
- ❖ For most languages, the size of the array or the number of elements it may hold is set when the array is declared.
- ❖ The size of the array cannot be changed (however, you can copy smaller arrays into larger arrays.) In addition, most programming languages require that all of the elements in the array have the same data type.
- ❖ In other words, if the array is of type number all of the elements must also be number values. Arrays are not restricted to just numbers and characters.
 - **Element:** An element is the name given to one of the values stored in the array. If the array is declare and store 5 values, it is said to have 5 elements.
 - **Index:** The index is an integer that uniquely identifies each element in the array. Indexes typically start at 0 and increment by one until the last element-1.

Reasons for Using Arrays:

- ❖ There are many reasons for using arrays in your programs. The four most compelling reasons for using arrays would be:
 1. Not knowing the number of variables needed until runtime.
 2. For a large number of variables that would be difficult to assign unique names.
 3. Requirement to implement a search or sort mechanism within your program logic.
 4. A need to improve program performance.

- ❖ In the following array, I have graphically represented five elements which each hold a number.
- ❖ For most programming languages, array elements start with a number zero and increase by one the indexes must be numeric and they must be integer best provide a unique address for each element.
- ❖ In other words, the index to the first item is [0] and the index to the last item is [number of elements - 1].
- ❖ Following figure show one dimensional array holding five values:



- ❖ We can use the same name to access the variable, but we need some way to differentiate between the individual slots.
- ❖ To this we use an *index* into the array, we can assign value to array directly, for example :

Array name variable

Index

elements

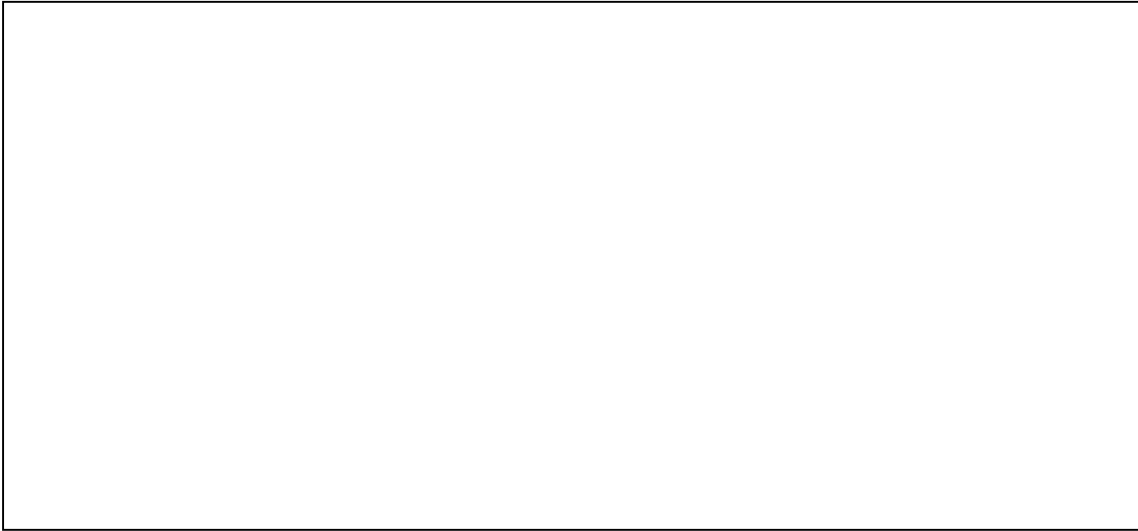
```

arrayName [0] = 90
arrayName [1] = 87
arrayName [2] = 97
arrayName [3] = 69
arrayName [4] = 77

```

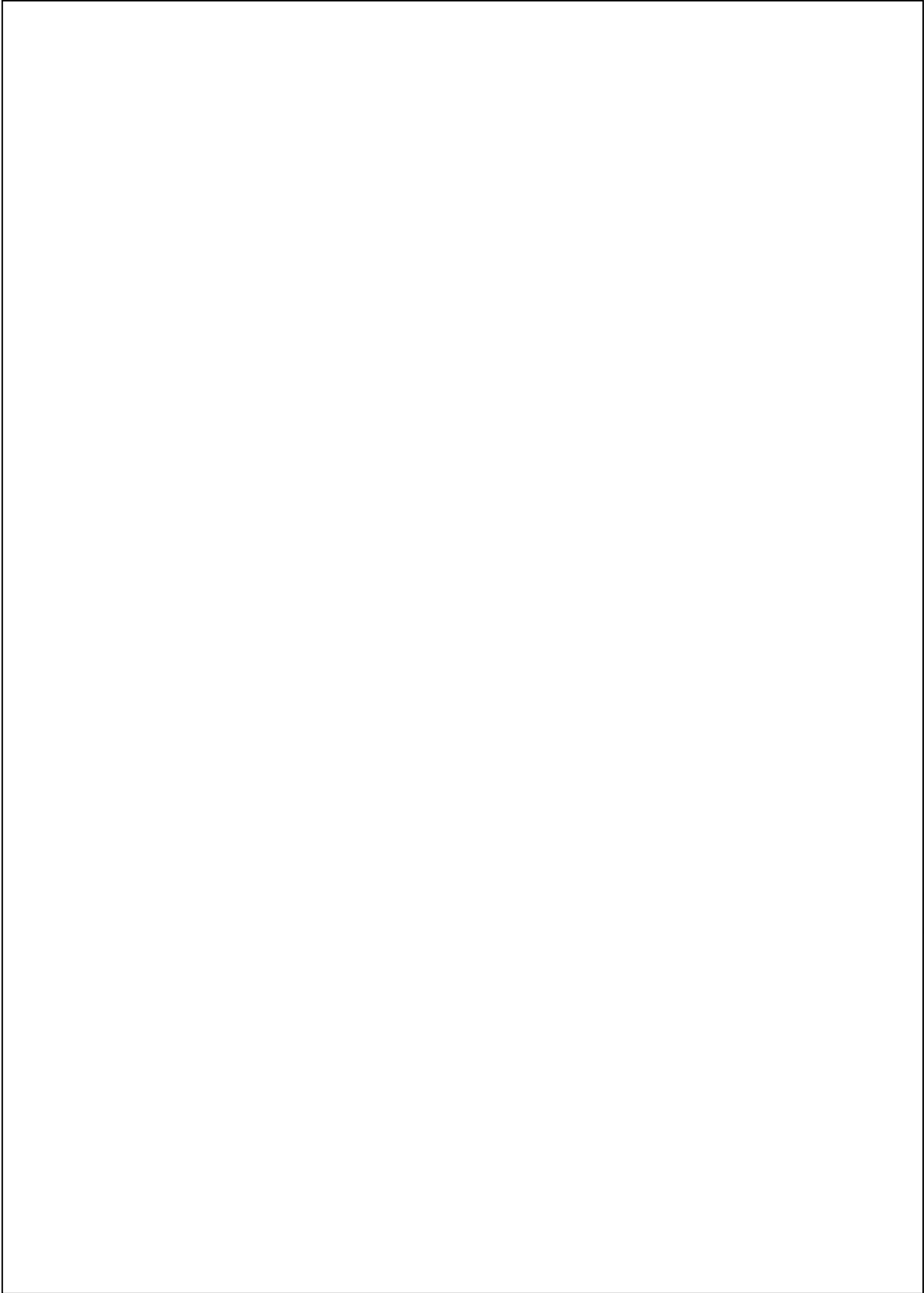
Example:

Write pseudo code to input array of seven elements and output it:

**Example:**

Write pseudo code and draw a flow chart to input 10 numbers in array, and output their square.





Example:

Write pseudo code and draw flowchart to display the even elements in the array of five elements.

Homework number (7) delivered next week

- 1) Write a pseudo code and draw a flowchart to output the summation one-dimensional array of six elements.
- 2) Write a pseudo code and draw a flowchart to output the one-dimensional array of ten elements in reverse order.
- 3) Write a pseudo code and draw a flowchart that add one-dimensional array (A) to one-dimensional array (B) in one-dimensional array(C), suppose the array A,B and C have size of nine elements.
- 4) Write a pseudo code and draw a flowchart that displays the maximum and minimal value in one-dimensional array that have size five elements.