جامـــــــعة المـــــستـقبـل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قـســـــــم الانظمة الطبية الذكية

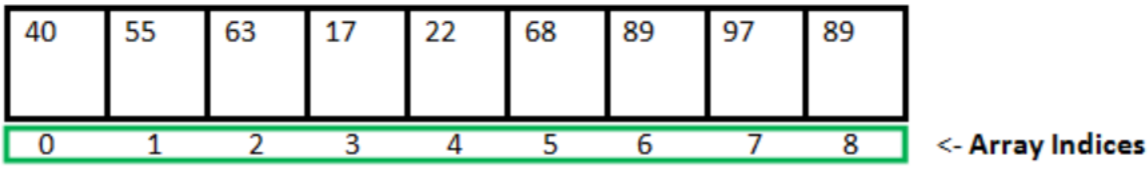
Lab: (2)

❖Arrays Part I

Subject: Computer Programming (I)
Level: First
Lecturer: Asst. Lect. Ali Saleem Haleem

Arrays in Java

- In Java, all arrays are dynamically allocated.
- Arrays may be stored in contiguous memory [consecutive memory locations].
- Since arrays are objects in Java, we can find their length using the object property length. This is different from C/C++, where we find length using sizeof.
- A Java array variable can also be declared like other variables with [] after the data type.
- The variables in the array are ordered, and each has an index beginning with 0.
- Java array can also be used as a static field, a local variable, or a method parameter.
- An array can contain primitives (int, char, etc.) and object (or non-primitive) references of a class depending on the definition of the array. In the case of primitive data types, the actual values might be stored in contiguous memory locations(JVM does not guarantee this behavior). In the case of class objects, the actual objects are stored in a heap segment.



Array Length = 9
First Index = 0
Last Index = 8

Creating, Initializing, and Accessing an Arrays

❖ One-Dimensional Arrays

The general form of a one-dimensional array declaration is:

-- type var-name[];
-- type[] var-name;

An array declaration has two components: the type and the name. type declares the element type of the array. The element type determines the data type of each element that comprises the array. Like an array of integers, we can also create an array of other primitive data types like char, float, double, etc., or user-defined data types (objects of a class). Thus, the element type for the array determines what type of data the array will hold.

Instantiating an Array

When an array is declared, only a reference of an array is created. To create or give memory to the array, you create an array like this: The general form of new as it applies to one-dimensional arrays appears as follows :

var-name = new type [size];

Example 1:

```
//declaring array
int intArray[];
// allocating memory to array
intArray = new int[20];
// combining both statements in one
int[] intArray = new int[20];
```

Array Literal

In a situation where the size of the array and variables of the array are already known, array literals can be used.

```
// Declaring array literal

int[] intArray = new int[]{ 1,2,3,4,5,6,7,8,9,10 };
```

Accessing Array Elements using for Loop

Each element in the array is accessed via its index. The index begins with 0 and ends at (total array size)-1. All the elements of array can be accessed using Java for Loop.

```
// accessing the elements of the specified array

for (int i = 0; i < arr.length; i++)

        System.out.println("Element at index " + i + " : "+ arr[i]);
```

Example 2:

```
// Java program to illustrate creating an array
// of integers,  puts some values in the array,
// and prints each value to standard output.

class GFG {
   public static void main(String[] args)
   {
      // declares an Array of integers.
      int[] arr;

      // allocating memory for 5 integers.
      arr = new int[5];

      // initialize the first elements of the array
      arr[0] = 10;

      // initialize the second elements of the array
```

```
        arr[1] = 20;

        // so on...
        arr[2] = 30;
        arr[3] = 40;
        arr[4] = 50;

        // accessing the elements of the specified array
        for (int i = 0; i < arr.length; i++)
            System.out.println("Element at index " + i
                               + " : " + arr[i]);
    }
}
```

Output

```
Element at index 0 : 10
Element at index 1 : 20
Element at index 2 : 30
Element at index 3 : 40
Element at index 4 : 50
```