Arrays

An array is the simplest data structure, a collection of related data items and a way of organizing data, in Java which is a container object used to store a fixed number of data. This collection of data must be of the same type such a list of numbers of type int or double or a list of strings.

	First index									
indices	0	1	2	3	4	5	6	7	8	9
elements										
	Array length is 10									

Example of an array of 10 elements

An array is used to hold a number of the same type of value instead of declaring individual variables such as score1, score2, score3, …, and score9 to represent individual value. Each item in an array is called an element which is accessed through its corresponding index. Since array is used to store a fixed number of data, a size of the array must be given when it is created.

Declaring and Initializing an array

For example, a simple array that can hold a collection of ten integers can be created as follows:

int score [ ] = new int [10];

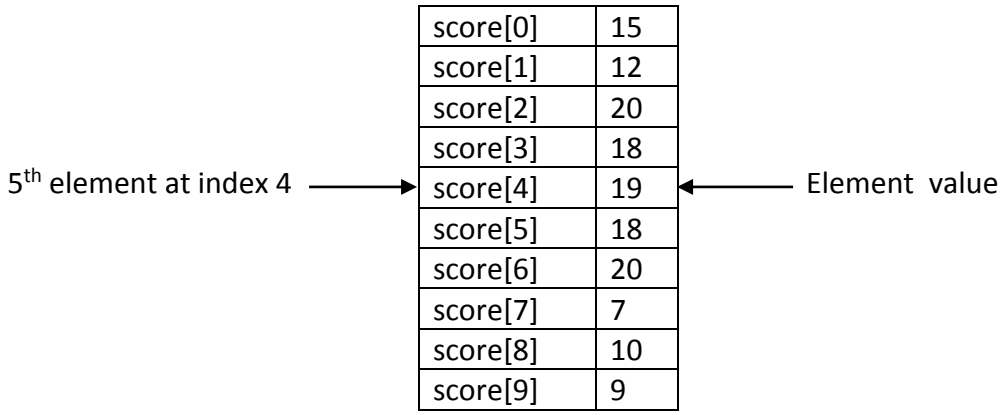
This Java statement declares and initializes an integer array. The square brackets [ ] on the left indicates that the int score variable is an array. The square brackets and an integer, [10], on the left indicate that the array has the maximum capacity of ten elements which can be referred by Java statement as follows:

score[0], score[1], score[2], score[3], score[4], score[5], score[6], score[8], score[9]

The following Java statements and figure illustrate the array with sample elements.

score[0] = 15;
score[1] = 12;
score[2] = 20;
score[3] = 18;
score[4] = 19;
score[5] = 18;
score[6] = 20;
score[7] = 7;

score[8] = 10;
score[9] = 9;



Accessing and Processing element values in an array

Each element of an array can be accessed directly by referring to the name of array with the index number of the item being access such as:

- **Scanner input = new Scanner(System.in);**
  System.out.print("Enter an integer for index 0:");
  score[0] = 15;
  This Java statement prompts the user to enter an integer for the first element in the array.
- **System.out.println(score[0]);**
  This Java statement displays the first score in the array.
- **int sum = score[0] + score[9];**
  This Java statement adds first and last scores and assigns the result to an int variable sum.

- **ArrayExample1.java**

```
1  public class ArrayExample1 {
2    public static void main(String [] args) {
3      // declare and initialize an integer array
4      int score [] = new int [5];
5      score[0] = 10;
6      score[2] = 20;
7      score[1] = 12;
8      score[3] = 4;
9      score[4] = 9;
10
11     // display an element of the array
12     System.out.printf("Element in index 0:%d", score[0]);
13     System.out.printf("\nElement in index 2: %d", score[2]);
14
15     // example of adding elements of index 0 and 2
16     int sum = score[0] + score[2];
17     System.out.printf("\n\nSum: of index 0 & 2 is %d", sum);
18
19      // display all elements in the array by the order of index
20     System.out.printf("\n\nDisplay all elements in array score");
21     for(int index = 0; index < 5; index++) {
22       System.out.printf("\n%d", score[index]);
23     }
24   }
25 }
```

To access all elements in an array, for loop is often used due to the following reasons.

- All of the elements in an array are of the same type.
- The size of array is always specified and known.
- For examples:
    - (Initializing arrays with Scanner) This loop initializes the array score by prompting the user to enter an integer for each element.

      Scanner input = new Scanner(System.in);
      for(int index = 0; index < 10, index++) {
      System.out.print(“Enter an integer:”);
      score[index] = input.nextInt();
      }
    - (Printing arrays) This loop displays all element values in the array by the order of index.

      for(int index = 0; index < 10, index++) {
      System.out.prinf(“%d\n”, score[index]);
      }

- (Summing all elements) Use a variable named total to store the sum of all elements. The variable total must be initialized to 0 to begin with. Each element will then be added to the variable total by the loop.

```
int total = 0;
for(int index = 0; index < 10, index++) {
    total += score[index];
}
```

- (Finding the smallest element) Use a variable named lowest to store the smallest element. The variable lowest must be initialized to the first element in the array to begin with. Each element will then be compared to the variable lowest by the loop. If an element in the array is smaller than the variable lowest, the if statement catches it and set the variable lowest to that particular element.

```
int lowest = score[0];
for(int index = 0; index < score.length; index++) {
    if(score[index] < lowest) {
       lowest = score[index];
     }
}
```

- (Counting elements) Use a variable named count to store a number of counting for a particular case. The variable count must be initialized to 0 to begin with. Each element will then be compared to a condition by the loop. If an element in the array is matched with the condition, the count variable is incremented by 1 (counting). For example, this loop counts the scores that are 10 and above.

```
int count = 0;
for(int index = 0; index < score.length; index++) {
    if(score[index] >= 10) {
       count++;
    }
}
```

- **ArrayExample2.java**

```java
import java.util.Scanner;
public class ArrayExample2 {
  public static void main(String [] args) {
     Scanner input = new Scanner(System.in);
     System.out.print("How many scores would you like to enter?:");
     //get the size of the score
     int size = input.nextInt();
     //create an integer score with the size entered by the user
     int score [] = new int [size];
     //use a for loop to assign a value to each index
     for(int index = 0; index < score.length; index++) {
        System.out.printf("Enter score %d", index+1);
        score[index] = input.nextInt();
     }
     //compute the total of the numbers in the score
     int total = 0;
     for(int index = 0; index < score.length; index++) {
       total += score[index];
     }
     System.out.println("Total score is: " + total);

     //find the smallest number in the score
     int lowest = score[0];
     for(int index = 0; index < score.length; index++) {
       if(score[index] < lowest) { //if number in score is less than smallest
         lowest = score[index]; //change the smallest to that number
       }
     }
     System.out.println("Lowest score = " + lowest);

     int count = 0;
     for(int index = 0; index < score.length; index++) {
       //count how many scroes is 10 and above
       if(score[index] >= 10) {
         count++;
       }
     }
     System.out.println("Number of passing scores which is 10 and above: " + count);
  }
}
```