ACPCE
Where knowledge is second nature

Jawahar Education Society's
A. C. Patil College of Engineering, Kharghar
Navi Mumbai 410210

Student Name: Chetan Ingale

Course Name: C.S.E. (IoT CS BC)

Year: S.E.

Roll No.: 17

PRN No.: 221111030

Course code: CSL304

Semester: 3

Experiment Evaluation Sheet

Experiment No.: 6

Experiment Name:
Program on array

Sr No.	Evaluation Criteria	Marks (Out of 9)	Performance Date	Correction Date and Signature of Instructor
1	Experiment Performance			
2	Journal Performance			
3	Punctuality			
Total				

Aim : Program on array

Software required : Java, Javac.

Theory :

Arrays in Java :-

Normally, an array is a collection of similar type of elements which has contiguous memory location.

Java array is an object which contains elements of a similar data type. Additionally, The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.

Unlike C/C++, we can get the length of the array using the length member. In C/C++, we need to use the sizeof operator.

In Java, array is an object of a dynamically generated class. Java array inherits the Object class, and implements the Serializable as well as Cloneable interfaces. We can store primitive values or objects in an array in Java. Like C/C++, we can also create single dimentional or multidimentional arrays in Java.

Moreover, Java provides the feature of anonymous arrays which is not available in C/C++.

The general form of a one-dimensional array declaration is

```
type var-name[];
OR
type[] var-name;
```

The general form of a two-dimensional array declaration is

```
int[][] twoD_arr = new int[10][20];
```

Code 6.a :

```
public class ReverseArray {
   public static void main(String[] args) {
      int[] arr = new int[] {1, 2, 3, 4, 5};

      System.out.print("Elements in Orignal Array : ");
      for(int i = 0; i < arr.length; i++) {
         System.out.print(arr[i] + " ");
      }
      System.out.println();
      System.out.print("Elements in Reverse Array : ");
      for(int i = arr.length - 1; i >= 0; i--) {
         System.out.print(arr[i] + " ");
      }
      System.out.println();
   }
}
```

```
chetan_i_007@HP-15-BS661TX:/media/chetan_i_007/Ventoy/College/OOPs/Exp06$ java ReverseArray
Elements in Orignal Array : 1 2 3 4 5
Elements in Reverse Array : 5 4 3 2 1
```

Code 6.b :

```
public class CopyArray{
   public static void main(String[] args) {
      int[] arr1 = new int[] {1, 2, 3, 4, 5};
      int arr2[] = new int[arr1.length];
      for(int i = 0; i < arr1.length; i++) {
         arr2[i] = arr1[i];
      }

      System.out.print("Elements in Orignal Array : ");
      for(int i = 0; i < arr1.length; i++) {
         System.out.print(arr1[i] + " ");
      }

      System.out.println();

      System.out.print("Elements in New Array : ");
      for(int i = 0; i < arr2.length; i++) {
         System.out.print(arr2[i] + " ");
      }

      System.out.println();
   }
}
```

Output 6.b :

```
chetan_i_007@HP-15-BS661TX:/media/chetan_i_007/Ventoy/College/OOPs/Exp06$ java CopyArray
Elements in Orignal Array : 1 2 3 4 5
Elements in New Array : 1 2 3 4 5
```

Conclusion :

With this experiments we learn how to implement arays in java programming language.

Output 4.b :

```
student@csiot-ThinkCentre-M70s:~/Chetan17/OOPs/Exp04$ javac overload02.java
student@csiot-ThinkCentre-M70s:~/Chetan17/OOPs/Exp04$ java overload02
Volume =2400.0
Volume =1000.0
Volume =-1.0
student@csiot-ThinkCentre-M70s:~/Chetan17/OOPs/Exp04$
```

Code 4.c :

```
class parameter {
    int a, b, c, add;

    void display () {
        System.out.println("No Parameters");
    }
    void display (int a, int b) {
        System.out.println("a = " + a + "\nb = " + b);
    }
    void display (double a) {
        System.out.println(" double a = " + a );
    }
}

class overload03 {
    public static void main(String args[]) {
        parameter mypara = new parameter();

        mypara.display();
        mypara.display(2, 5);
        mypara.display(2);
        mypara.display(25.63);
    }
}
```

Output 4.c :

```
student@csiot-ThinkCentre-M70s:~/Chetan17/OOPs/Exp04$ javac overload03.java
student@csiot-ThinkCentre-M70s:~/Chetan17/OOPs/Exp04$ java overload03
No Parameters
a = 2
b = 5
 double a = 2.0
 double a = 25.63
student@csiot-ThinkCentre-M70s:~/Chetan17/OOPs/Exp04$
```

Conclusion :

With this programs we learn how to implement method and constructor overloading in java programming language.