

PRACTICAL 3

ARRAYS AND LOOPS AND CONDITIONAL STATEMENTS

ARRAYS:

- ➔ An array in Java is a group of like-typed variables referred to by a common name. Arrays in Java work differently than they do in C/C++. Following are some important points about Java arrays.
- ➔ In Java, all arrays are dynamically allocated.
- ➔ 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.
- ➔ 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.
- ➔ The variables in the array are ordered, and each has an index beginning from 0.
- ➔ Java array can be also be used as a static field, a local variable, or a method parameter.
- ➔ The size of an array must be specified by int or short value and not long.
- ➔ The direct superclass of an array type is Object.
- ➔ 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 are stored in contiguous memory locations. In the case of class objects, the actual objects are stored in a heap segment.

LOOPS:

- ➔ The Java for loop is used to iterate a part of the program several times. If the number of iterations is fixed, it is recommended to use for loop.
- ➔ There are three types of loop in java:
 - For loop: The java for loop is used to iterate a part of the program several times. If the number of iterations is fixed, it is recommended to use for loop.
 - `for(initialization; condition; increment/decrement){`
 - `//statement or code to be executed`
 - `}`
 - While loop: The Java while loop is used to iterate a part of the program several times. If the number of iterations is not fixed it is recommended to use while loop.
 - `while (condition) {`
 - `// code block to be executed`
 - `}`
 - Do-while loop: The Java do-while loop is to iterate a part of the program several times. Use it if the number of iterations is not fixed and you must have to execute the loop at least once.
 - `do{`
 - `//code to be executed / loop body`
 - `//update statement`
 - `}while (condition);`

CONDITIONAL STATEMENTS:

➔ Java supports the usual logical conditions.

➔ Java has the following conditional statements:

- Use if to specify a block of code to be executed, if a specified condition is true
 - `if (condition) {`
 - `// block of code to be executed if the condition is true`
 - `}`
- Use else to specify a block of code to be executed, if the same condition is false
 - `if (condition) {`
 - `// block of code to be executed if the condition is true`
 - `} else {`
 - `// block of code to be executed if the condition is false`
 - `}`
- Use else if to specify a new condition to test, if the first condition is false
 - `if (condition1) {`
 - `// block of code to be executed if condition1 is true`
 - `} else if (condition2) {`
 - `// block of code to be executed if the condition1 is false and condition2 is true`
 - `} else {`
 - `// block of code to be executed if the condition1 is false and condition2 is false`
 - `}`

Q1. Write a java program to Calculate the sum of all elements of an array.

```
package practical3;

import java.util.Scanner;

public class sum_of_all {
    public static void main(String[] args) {
        // take input for array
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter 10 numbers: ");
        int[] arr = new int[10];
        for (int i = 0; i < 10; i++) {
            arr[i] = scan.nextInt();
        }

        // calculate sum
        int sum = 0;
        for (int i = 0; i < 10; i++) {
            sum = sum + arr[i];
        }

        // print sum
        System.out.println("Sum of all the entered number is " +
sum);
    }
}
```

run:

Enter 10 numbers: 2 3 4 5 2 3 7 6 5 8

Sum of all the entered number is 45

BUILD SUCCESSFUL (total time: 14 seconds)

Q2. Write a program in java that takes a number as input and prints its multiplication table upto 10.

```
package practical3;

import java.util.Scanner;

public class mul_table {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Which multiplication table do you
want: ");
        int multiplicand = scan.nextInt();
        for (int multiplier = 0; multiplier <= 10; multiplier++) {
            System.out.println(multiplicand + " x " + multiplier +
" = " + (multiplicand * multiplier));
        }
    }
}
```

run:

Which multiplication table do you want:

5

5 x 0 = 0

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

BUILD SUCCESSFUL (total time: 3 seconds)

Q3. Write a java program to accept angles of a triangle and display equilateral, isosceles and scalene.

```
package practical3;

import java.util.Scanner;

public class display_triangle_type {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter lengths of side of a triangle: ");
        int a = scan.nextInt();
        int b = scan.nextInt();
        int c = scan.nextInt();

        if (a == b && b == c)
            System.out.println("It is an equilateral triangle");
        else if (a == b || b == c || a == c)
            System.out.println("It is an isosceles triangle");
        else
            System.out.println("it is a scalene triangle");
    }
}
```

run:

```
Enter lengths of side of a triangle: 4 5 6
it is a scalene triangle
BUILD SUCCESSFUL (total time: 3 seconds)
```

run:

```
Enter lengths of side of a triangle: 4 4 5
It is an isosceles triangle
BUILD SUCCESSFUL (total time: 2 seconds)
```

run:

```
Enter lengths of side of a triangle: 4 4 4
It is an equilateral triangle
BUILD SUCCESSFUL (total time: 2 seconds)
```

Q4. Write a java program to count the number of occurrences of given number in an array of integers.

```
package practical3;

import java.util.Scanner;

public class occur_in_array {
    public static void main(String[] args) {
        int arr[] = { 1, 2, 3, 4, 4, 5, 2, 2, 3, 7, 6, 5, 4, 4, 3,
2, 2, 1, 1, 4, 5, 3, 8, 8, 9, 6, 7, 2 };
        int arr_len = arr.length;
        Scanner scan = new Scanner(System.in);
        System.out.println("Which number do you want to count: ");
        int arr_find = scan.nextInt();

        int count = 0;
        for (int i = 0; i < arr_len; i++)
            if (arr_find == arr[i])
                count++;

        System.out.println("The number occurs " + count + "
times");
    }
}
```

run:

Which number do you want to count:

2

The number occurs 6 times

BUILD SUCCESSFUL (total time: 3 seconds)

Q5. Write a java program to accept 3x3 matrix and display the transpose of a given matrix.

```
package practical3;

public class transpose_of_matrix {
    public static void main(String[] args) {
        int matrix[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8,
9 } };
        int transpose[][] = new int[3][3];

        // printing original matrix
        System.out.println("The matrix is: ");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(matrix[i][j] + " ");
            }
            System.out.println();
        }

        // print transpose of matrix
        System.out.println("The transpose of the matrix is: ");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(matrix[j][i] + " ");
            }
            System.out.println();
        }
    }
}
```

```
run:
The matrix is:
1 2 3
4 5 6
7 8 9
The transpose of the matrix is:
1 4 7
2 5 8
3 6 9
BUILD SUCCESSFUL (total time: 0 seconds)
```