Lab Manual

**OOP**

**Laboratory 06:**

**Statement Purpose:**

At the end of this lab, the students should be able to:

* Implement Arrays in Java
* Implement Control Structures in Java

• Use decision control structures (if, else, switch) which allows selection of specific sections of code to be executed

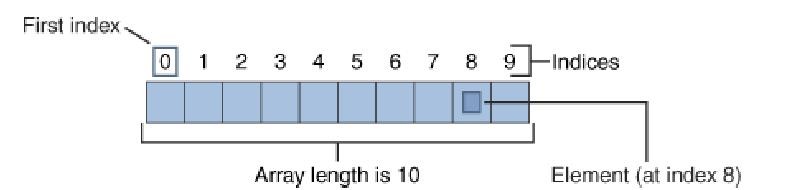
• Use repetition control structures (while, do-while, for) which allow executing specific sections of code a number of times

• Use branching statements (break, continue, return) which allows redirection of program flow

**Arrays:**

Arrays are special data types that let us store specified number of variables from the same type using one variable name.

Arrays are an indexed data type, which means they are storing different elements discriminating between them using unique index for each one. Figure shows the structure of an array.



As Figure shows, the size of an array is fixed; we will refer to array maximum size as array length, it is also clear that indices of an array are zero-based, that is, they start from 0 to length –1; For example, the array shown in Figure has a length of 10 (stores up to 10 elements), and the last index is 9.

**One-Dimensional Arrays in Java**

Declaring and using an array in Java is similar to declaration of any variable; you have to specify

data type and name, in addition to this, you have also to specify the length of the array before using it. Array definition format is as follows:

**DataType[] arrayName;**

To initialize the array, use new keyword and specify the length.

**arrayName = new DataType[length];**

Here are some examples of array declaration:

**int[] a = new int[6]; //Array of integers of length 6**

**String[] b = new String[10]; //Array of strings of length 10**

To access array members, use array name with desired index specified between square brackets [].

**int[] x = new int[3];**

**x[0] = 5; //Store 5 in the first index**

**x[1] = 10; //Store 10 in the second index**

**x[2] = 15; //Store 15 in the third index**

**System.out.println(x[2] + x[1]); //Prints 25**

If you try to access an index out of the range of array (greater or equals array length) JVM will throw an IndexOutOfRangeException.

**int[] x = new int[3]; //Valid indices are 0, 1 and 2**

**x[3] = 7; //IndexOutOfRangeException**

Another technique to declare an array is to directly provide its members between brackets {}, separated using commas ','.

**int[] x = {10, 15, 20, 25, 100}; //Length = 5**

**System.out.println(x[3]); //Prints 25**

**String[] days = {"Sat", "Sun", "Mon", "Tue", "Wed", "Thu", "Fri"};**

**System.out.println("We are leaving in " + days[2]); //We are leaving on Monday**

**Multidimensional Arrays (Arrays of Arrays)**

Java also provides arrays of arrays, this means we create an array, and each element in that array is an array itself. These arrays are also called multidimensional arrays. The declaration of multidimensional arrays is shown below:

**DataType[][] arrayName;**

**arrayName = new DataType[rows][columns]**

**Control Structures:**

A control structure allows us to change the ordering of how the statements in our programs are executed. There are two types of Control Structures

* Decision control structures :Allows to select specific sections of code to be executed
* Repetition control structures :Allows to execute specific sections of the code a number of times

**Decision Control Structures:**

A Java statement that allows us to select and execute specific blocks of code while skipping other sections.

* if-statement
* if-else-statement
* if-else if-statement
* switch statement

**Example: 1**

//Demonstrate if-else-if statements.

**public** **class** ifelse {

**public** **static** **void** main(String[] args) {

**int** month=4;

String season;

**if**(month==12 || month==1 || month==2)

season="winter";

**else** **if**(month==3 || month==4 || month==5)

season="spring";

**else** **if**(month==6 || month==7 || month==8)

season="summer";

**else** **if**(month==9 || month==10 || month==11)

season="autumn";

**else**

season="Invalid Month";

System.*out*.println("April is in the " + season + ".");

}

}

**Output:**

# 

**Example: 2**

//A simple example of the switch

**public** **class** switchex {

**public** **static** **void** main(String[] args) {

**for**(**int** i=0;i<6;i++)

**switch**(i){

**case** 0:

System.*out*.println("i is zero.");

**break**;

**case** 1:

System.*out*.println("i is one.");

**break**;

**case** 2:

System.*out*.println("i is two.");

**break**;

**case** 3:

System.*out*.println("i is three.");

**break**;

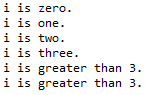
**default**:

System.*out*.println("i is greater than 3.");

}

}

**Output:**



**Repetition Control Structures:**

Repetition control structures are a Java statement that allows us to execute specific blocks of code a number of times.

* while-loop
* do-while loop
* for-loop

**Example: 1**

//demonstrate the while loop

**public** **class** whileex {

**public** **static** **void** main(String[] args) {

**int** n=10;

**while**(n>0){

System.*out*.println("tick" +n);

n--;

}

}

}

**Output:**



**Example: 2**

//demonstrate the do-while loop

**public** **class** dowhileex {

**public** **static** **void** main(String[] args) {

**int** n=10;

**do**{

System.*out*.println("tick" +n);

n--;

}**while**(n>0);

//the above loop can be more efficiently written as follows:

/\*

do{

System.out.println("tick" +n);

}while(--n > 0);

\*/

}

}

**Output:**



**Example: 3**

//demonstrate for loop (loop control variable is declared inside the for)

**public** **class** forex {

**public** **static** **void** main(String[] args) {

**for** (**int** n=10;n>0;n--)

System.*out*.println("tick" +n);

}

}

**Output:**



**Branching Statements**

Branching statements allows us to redirect the flow of program execution. Java offers three branching statements:

* break
* continue
* return.

**Example: 1**

//Using break with nested loops

**public** **class** breaknested {

**public** **static** **void** main(String[] args) {

**for**(**int** i=0;i<3;i++){

System.*out*.print("Pass" +i+ ":");

**for**(**int** j=0;j<100;j++){

**if**(j==10) **break**;

System.*out*.print(j + " ");

}

System.*out*.println();

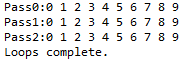
}

System.*out*.println("Loops complete. ");

}

}

**Output:**



**Example: 2**

//Demonstrate continue

**public** **class** continueex {

**public** **static** **void** main(String[] args) {

**for**(**int** i=0;i<10;i++){

System.*out*.print(i + " ");

**if**(i%2 == 0)**continue**;

System.*out*.println("");

}

}

}

**Output:**



**Example: 3**

//Demonstrate return

**import** java.util.Scanner;

**public** **class** returnex {

**public** **static** **void** main(String[] args) {

Scanner data = **new** Scanner(System.*in*);

**int** num1, num2;

System.*out*.println("Enter 1st number");

num1 = data.nextInt();

System.*out*.println("Enter 2nd number");

num2 = data.nextInt();

data.close();

*total*(num1,num2);

}

**public** **static** **void** total(**int** numx, **int** numy){

**int** sum=0;

sum=numx+numy;

**if**(sum==0)

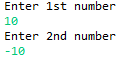
**return**;

System.*out*.println("Sum of 2 number:"+sum);

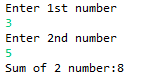
}

}

**Output:**



**Output:**



**Lab Tasks:**

**Task 1: Write a Java application that uses looping to print the following table of values:**

### 

**Marks: 3**

**Task 2: Write a program using while loop which calculate square of every number and then sum the squares of all numbers. Output should be like as shown below: Marks: 2**

**Output:**

1

4

9

16

25

36

49

64

81

100

Total is 385

**Task 3: What does the following program print? Mark: 1**

**public** **class** Task3 {

**public** **static** **void** main(String[] args) {

**int** count = 1;

**while** ( count <= 10 ) {

System.*out*.println( count % 2 == 1 ? "\*\*\*\*" : "++++++++" );

++count;

}

} // end main

} // end class Task3

**Task 4: Determine the output for each of the given sets of code when x is 9 and y is 11 and when x is 11 and y is 9. Note that the compiler ignores the indentation in a Java program. Also, the Java compiler always associates an else with the immediately preceding if unless told to do otherwise by the placement of braces ({}). Marks: 2**

a) **if** ( x < 10 )

**if** ( y > 10 )

System.*out*.println( "\*\*\*\*\*" );

**else**

System.*out*.println( "#####" );

System.*out*.println( "$$$$$" );

b) **if** ( x < 10 ) {

**if** ( y > 10 )

System.*out*.println( "\*\*\*\*\*" );

}

**else** {

System.*out*.println( "#####" );

System.*out*.println( "$$$$$" );

}

**Task 5: Write a test program that find out the minimum value from an array of 10 digits.**

**Marks: 2**