

Lab-04

Java Control Statement, Operators And Arrays.

Objectives:

Understanding the control statements of Java including Loops & if-else. Exploring different operators used in Java and Arrays, array index, single and multi-dimensional arrays, traversing the array using loop

Theory:

Iteration Statements

A loop can be used to tell a program to execute statements repeatedly. Java provides a powerful construct called a loop that controls how many times an operation or a sequence of operations is performed in succession. A loop can be nested inside another loop. Different form of loops can be nested with one another.

Java provides three types of loop statements:

1. while loops,
2. do-while loops, and
3. for loops.

Loop Syntax

<i>while</i>	<i>do-while</i>	<i>for</i>
<pre>while (loop-continuation- condition) { // Loop body Statement(s); }</pre>	<pre>do { // Loop body; Statement(s); } while (loop-continuation-condition);</pre>	<pre>for (initial-action; loop-continuation-condition; action-after-each-iteration) { // Loop body; Statement(s); }</pre>

Selection Statements

Selection Statements of Java programming language decides the next statement for execution. Selection statements use conditions that are Boolean expressions. A Boolean expression is an expression that evaluates to a Boolean value: true or false. An if statement can be inside another if statement to form a nested if statement.

Java provides three types of selection statements:

1. If statements
2. If-else statements
3. If-else if
4. Switch

if statement executes the statements if the condition is true. if-else statement is two way statement that decides which statements to execute based on whether the condition is true or false. Multi-way **if-else** statement is the preferred coding style for multiple alternative if statements.

If-else Syntax

<i>if</i>	<i>do-while</i>	<i>for</i>
<pre>if (boolean-expression) { statement(s); }</pre>	<pre>if (boolean-expression) { //for-the-true-case; } else { //for-the-false-case; }</pre>	<pre>if (boolean-expression) { // for-1st-case; } else if(boolean-expression) { // for-2nd-case; } else { //for-default-case; }</pre>

Switch Syntax

```
switch (expression) {
case value1:
    // statement sequence
    break;
case value2:
    // statement sequence
    break;
.
.
.
casevalueN:
    // statement sequence
    break;
default:
    // default statement sequence
}
```

Operators in Java

There are many operators provide by Java. Following are the most commonly used operators.

Comparison Operators:

Comparison operators can be used to compare two values. The result of the comparison is a Boolean value: true or false. Java provides six comparison operators.

Comparison Operators		
Java Operator	Mathematics Symbol	Name
<	<	less than
<=	≤	less than or equal to
>	>	greater than
>=	≥	greater than or equal to
==	=	equal to
!=	≠	not equal to

Figure 2.3: Comparison Operators

Logical Operators:

The logical can be used to create a compound Boolean expression. Sometimes, whether a statement is executed is determined by a combination of several conditions.

Boolean Operators		
Operator	Name	Description
!	not	logical negation
&&	and	logical conjunction
	or	logical disjunction
^	exclusive or	logical exclusion

Figure 2.3: Boolean Operators

Array

A single array variable can reference a large collection of data. Java and most other high-level languages provide a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. Once an array is created, its size is fixed. An array reference variable is used to access the elements in an array using an index

Syntax:

```
elementType[] arrayRefVar = new elementType[arraySize]; // 1d array
elementType[][] arrayRefVar;                             // 2d array
```

To assign values to the elements, use the syntax:

```
arrayRefVar[index] = value;           // 1d array
arrayRefVar[row][column] = value;     // 2d array
```

Lab Task:

1. Design a Java program to use an if-else-if ladder to determine which season a particular month is in.
2. Design a Java program to generate following pattern

.....

....

...

..

.

3. Design a program using one-dimensional array of numbers and finds the average of a set of numbers.

Lab Assignment:

- 1 Design a program 1 of this lab using *switch* statement.
- 2 The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, every subsequent value is the sum of the 2 values preceding it. Design a Java Program that print the nth value of the Fibonacci sequence?
- 3 Design a Java program that prompts the user for an integer and then prints out all the prime numbers up to that Integer?
- 4 Design a Java program for sorting a given array of numbers in ascending order and descending order?
- 5 Design a Java program to multiply two given matrices using 2D array.

Conclusion:



FACULTY OF ENGINEERING SCIENCES AND TECHNOLOGY
