

## Control Statements in Java

### Aim:

To understand and implement decision-making and looping control statements in Java.

### PRE LAB EXERCISE

#### QUESTIONS

- ✓ List different control statements in Java.

**Control statements are used to control the flow of execution of a program.**

**Java control statements are classified into selection, iteration, and jump statements.**

**Selection statements include if, if-else, and switch.**

**Iteration statements include for, while, and do-while.**

**Jump statements include break, continue, and return.**

- ✓ Difference between for, while, and do-while loops.

**The for loop is used when the number of iterations is known beforehand.**

**The while loop checks the condition before executing the loop body.**

**The do-while loop executes the loop body at least once.**

**In do-while, the condition is checked after the loop body.**

- ✓ What is the use of break and continue?

**The break statement is used to terminate a loop or switch statement.**

**It transfers control to the statement following the loop.**

**The continue statement skips the remaining statements of the current iteration.**

**After continue, the loop proceeds with the next iteration.**

### IN LAB EXERCISE

#### Objective:

To implement if-else and looping statements.

#### INPUT STATEMENT:

#### SCANNER CLASS

- ✓ The Scanner class in Java is used to read input from the user through the keyboard.
- It is available in the package java.util.
- ✓ The Scanner object reads different types of input such as integer, float, double, and string and stores them in variables.
- ✓ To use the Scanner class, it must be imported before using it in the program.

#### SYNTAX:

- ✓ Scanner sc = new Scanner(System.in);

### Commonly Used Scanner Methods:

- ✓ nextInt() – reads an integer value
- ✓ nextFloat() – reads a float value
- ✓ nextDouble() – reads a double value
- ✓ next() – reads a single word
- ✓ nextLine() – reads a complete line of text

## PROGRAMS:

### Program 1: Check Whether a Number is Positive

```
class PositiveNumber {  
    public static void main(String[] args) {  
        int n = 12;  
        if (n > 0) {  
            System.out.println("Positive Number");  
        }  
    }  
}
```

### Output:

Positive Number

The screenshot shows the Eclipse IDE interface. At the top, there are two tabs: "main.java" and "PositiveNumber.java X". The "PositiveNumber.java" tab is active. Below the tabs is the Java code for PositiveNumber.java. The code defines a class PositiveNumber with a main method that prints "Positive Number" to the console if n is greater than 0. The code is numbered from 1 to 12. In the bottom right corner of the code editor, there is a small blue icon with a question mark. At the bottom of the screen is the Eclipse toolbar with icons for Problems, Javadoc, Declaration, Console, and Help. The "Console" tab is selected. The console window shows the output: "<terminated> PositiveNumber [Java Application] D:\eclipse\plugins\Positive Number|".

```
1 package programming.java.com;  
2  
3 class PositiveNumber {  
4     public static void main(String[] args) {  
5         int n = 12;  
6         if (n > 0) {  
7             System.out.println("Positive Number");  
8         }  
9     }  
10 }  
11  
12
```

## Program 2: Check Whether a Number is Even or Odd

```
class EvenOdd {  
    public static void main(String[] args) {  
        int n = 6;  
        if (n % 2 == 0)  
            System.out.println("Even Number");  
        else  
            System.out.println("Odd Number");  
    }  
}
```

### Output:

Even Number

The screenshot shows the Eclipse IDE interface. At the top, there are three tabs: 'main.java', 'PositiveNumber.java', and 'EvenOdd.java'. The 'EvenOdd.java' tab is active. Below the tabs, the code for 'EvenOdd.java' is displayed in a code editor window. The code checks if the number 12 is even or odd and prints the result. The code editor has line numbers from 1 to 13 on the left. In the bottom right corner of the code editor, there is a small blue icon with a white question mark. At the bottom of the screen, there is a toolbar with icons for 'Problems', 'Javadoc', 'Declaration', 'Console', and other options. The 'Console' tab is selected. The console window shows the output: '<terminated> EvenOdd [Java Application] D:\eclipse\plugins\org.e'. Below the console window, the text 'Even Number' is visible.

```
1 package programming.java.com;  
2  
3 class EvenOdd {  
4     public static void main(String[] args) {  
5         int n = 12;  
6         if (n % 2 != 0)  
7             System.out.println("Odd Number");  
8         else  
9             System.out.println("Even Number");  
10    }  
11 }  
12  
13
```

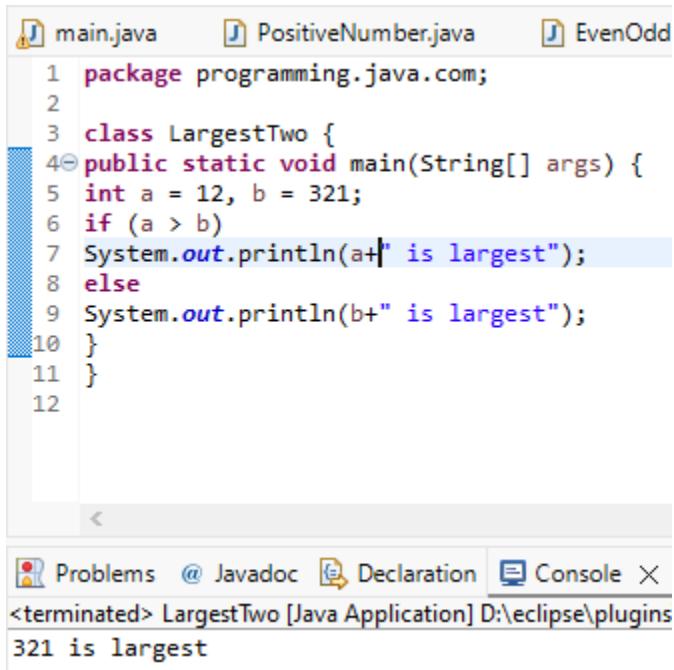
## Program 3: Find Largest of Two Numbers

```
class LargestTwo {  
    public static void main(String[] args) {  
        int a = 12, b = 321;
```

```
if (a > b)
System.out.println("A is largest");
else
System.out.println("B is largest");
}
}
```

### Output:

321 is largest



The screenshot shows the Eclipse IDE interface. At the top, there are three tabs: main.java, PositiveNumber.java, and EvenOdd. The main.java tab is active, displaying the following Java code:

```
1 package programming.java.com;
2
3 class LargestTwo {
4     public static void main(String[] args) {
5         int a = 12, b = 321;
6         if (a > b)
7             System.out.println(a+" is largest");
8         else
9             System.out.println(b+" is largest");
10    }
11 }
```

Below the code editor is the Eclipse status bar, which shows the current project is 'LargestTwo [Java Application]' and the file path is 'D:\eclipse\plugins'. The status bar also displays the output '321 is largest'.

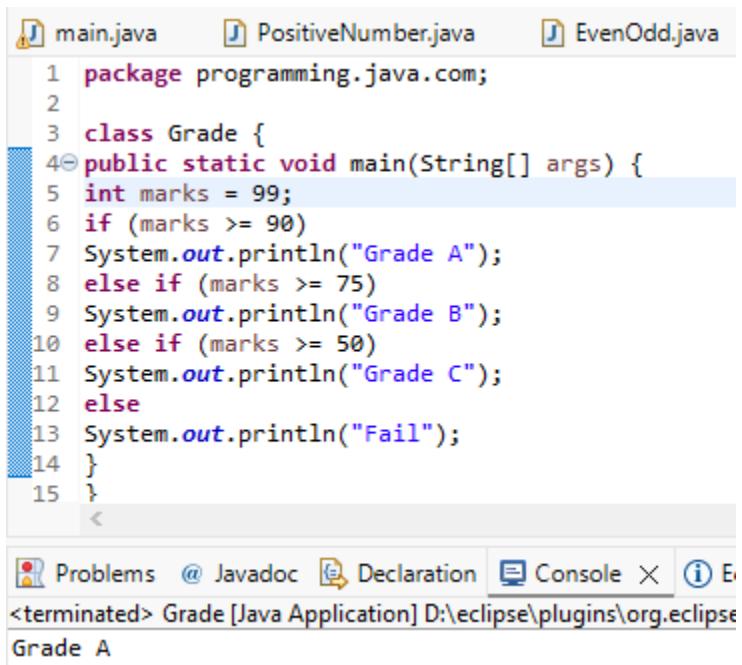
### Program 4: Grade Calculation

```
class Grade {
public static void main(String[] args) {
int marks = 99;
if (marks >= 90)
System.out.println("Grade A");
else if (marks >= 75)
System.out.println("Grade B");
else if (marks >= 50)
```

```
System.out.println("Grade C");
else
System.out.println("Fail");
}
}
```

### Output:

Grade A



The screenshot shows the Eclipse IDE interface. In the top bar, there are three tabs: main.java, PositiveNumber.java, and EvenOdd.java. The main.java tab is active and contains the following Java code:

```
1 package programming.java.com;
2
3 class Grade {
4     public static void main(String[] args) {
5         int marks = 99;
6         if (marks >= 90)
7             System.out.println("Grade A");
8         else if (marks >= 75)
9             System.out.println("Grade B");
10        else if (marks >= 50)
11            System.out.println("Grade C");
12        else
13            System.out.println("Fail");
14    }
15 }
```

Below the code editor, the Eclipse interface includes tabs for Problems, Javadoc, Declaration, Console, and others. The Console tab is active, showing the output of the program: "Grade A". The status bar at the bottom indicates the application is a Java Application running in the D:\eclipse\plugins\org.eclipse.

### Program 5: Day of the Week

```
class DaySwitch {
public static void main(String[] args) {
int day = 7;
switch (day) {
case 1: System.out.println("Monday"); break;
case 2: System.out.println("Tuesday"); break;
case 3: System.out.println("Wednesday"); break;
case 4: System.out.println("Thursday"); break;
case 5: System.out.println("Friday"); break;
```

```

case 6: System.out.println("Saturday"); break;
case 7: System.out.println("Sunday"); break;
default: System.out.println("Invalid Day");
}
}
}

```

### **Output:**

Wednesday

The screenshot shows the Eclipse IDE interface. In the top bar, there are three tabs: "PositiveNum...", "EvenOdd.java", and "LargestTwo.java". The main editor window displays the following Java code:

```

1 package programming.java.com;
2
3 class DaySwitch {
4     public static void main(String[] args) {
5         int day = 7;
6         switch (day) {
7             case 1: System.out.println("Monday"); break;
8             case 2: System.out.println("Tuesday"); break;
9             case 3: System.out.println("Wednesday"); break;
10            case 4: System.out.println("Thursday"); break;
11            case 5: System.out.println("Friday"); break;
12            case 6: System.out.println("Saturday"); break;
13            case 7: System.out.println("Sunday"); break;
14        default: System.out.println("Invalid Day");
15    }
}

```

Below the editor, the "Console" tab is selected, showing the output of the program:

```

Problems @ Javadoc Declaration Console <terminated> DaySwitch [Java Application] D:\eclipse\plugins\org.ecl
Sunday

```

### **Program 6: Print Numbers from 1 to 5**

```

class ForLoop {

public static void main(String[] args) {
    for (int i = 1; i <= 5; i++) {
        System.out.println(i);
    }
}
}

```

### Output:

```
1  
2  
3  
4  
5
```

The screenshot shows the Eclipse IDE interface. In the top bar, there are three tabs: EvenOdd.java, LargestTwo.java, and Grade.java. The main editor window displays a Java program named ForLoop. The code is as follows:

```
1 package programming.java.com;  
2  
3 class ForLoop {  
4     public static void main(String[] args) {  
5         for (int i = 1; i <= 5; i++) {  
6             System.out.println(i);  
7         }  
8     }  
9 }  
10  
11
```

The line `System.out.println(i);` is highlighted with a light blue background. Below the editor, the Eclipse tool bar has icons for Problems, Javadoc, Declaration, and Console. The Console tab is selected, showing the output of the application:

```
<terminated> ForLoop [Java Application] D:\eclipse\plugins\c  
1  
2  
3  
4  
5
```

### Program 7: Print Numbers from 1 to 5

```
class WhileLoop {  
    public static void main(String[] args) {  
        int i = 1;  
        while (i <= 5) {  
            System.out.println(i);  
            i++;  
        }  
    }  
}
```

### **Output:**

```
1  
2  
3  
4  
5
```

The screenshot shows the Eclipse IDE interface. In the top left, there are three tabs: 'LargestTwo.java', 'Grade.java', and 'DaySwitch.java'. Below them is a code editor window containing the following Java code:

```
1 package programming.java.com;  
2  
3 class WhileLoop {  
4     public static void main(String[] args) {  
5         int i = 1;  
6         while (i <= 5) {  
7             System.out.println(i);  
8             i++;  
9         }  
10    }  
11 }  
12
```

Below the code editor is a toolbar with icons for 'Problems', 'Javadoc', 'Declaration', 'Console', and a close button. Underneath the toolbar, the status bar displays the text '<terminated> WhileLoop [Java Application] D:\eclipse\plugins\'.

In the bottom left corner of the code editor, there is a vertical scroll bar. The scroll bar has several blue square markers, with one prominent blue square marker at the top and another at the bottom, indicating the current position of the scroll bar.

The 'Console' tab is selected, showing the output of the program:

```
1  
2  
3  
4  
5
```

### **Program 8: Print Numbers from 1 to 5**

```
class DoWhileLoop {  
public static void main(String[] args) {  
int i = 1;  
do {  
    System.out.println(i);  
    i++;  
} while (i <= 5);  
}
```

### **Output:**

```
1  
2  
3  
4  
5
```

The screenshot shows the Eclipse IDE interface. In the top left, there are three tabs: Grade.java, DaySwitch.java, and ForLoop.java. The Grade.java tab is currently active. Below the tabs is a code editor window containing the following Java code:

```
1 package programming.java.com;  
2  
3 class DoWhileLoop {  
4     public static void main(String[] args) {  
5         int i = 1;  
6         do {  
7             System.out.println(i);  
8             i++;  
9         } while (i <= 5);  
10    }  
11 }  
12 }
```

Below the code editor is a toolbar with icons for Problems, Javadoc, Declaration, and Console. The 'Console' tab is selected. The status bar at the bottom shows the application name 'DoWhileLoop [Java Application]' and the path 'D:\eclipse\plu'. The console output area displays the numbers 1 through 5, each on a new line.

```
1  
2  
3  
4  
5
```

### Program 9: Sum of First 5 Natural Numbers

```
class SumNumbers {  
    public static void main(String[] args) {  
        int sum = 0;  
        for (int i = 1; i <= 5; i++) {  
            sum = sum + i;  
        }  
        System.out.println("Sum = " + sum);  
    }  
}
```

```
}
```

### Output:

Sum = 15

The screenshot shows the Eclipse IDE interface. At the top, there are three tabs: DaySwitch.java, ForLoop.java, and WhileLoop.java. Below the tabs, the code editor displays the following Java program:

```
1 package programming.java.com;
2
3 class SumNumbers {
4     public static void main(String[] args) {
5         int sum = 0;
6         for (int i = 1; i <= 5; i++) {
7             sum = sum + i;
8         }
9         System.out.println("Sum = " + sum);
10    }
11 }
12
```

Below the code editor is the Eclipse perspective navigation bar with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is selected, showing the output of the program:

```
<terminated> SumNumbers [Java Application] D:\eclipse\plu
Sum = 15
```

### Program 10: Multiplication Table of a Number

```
class MultiplicationTable {
    public static void main(String[] args) {
        int n = 12;
        for (int i = 1; i <= 10; i++) {
            System.out.println(n + " x " + i + " = " + (n * i));
        }
    }
}
```

### Output:

```
12 x 1 = 12
12 x 2 = 24
12 x 3 = 36
12 x 4 = 48
12 x 5 = 60
12 x 6 = 72
12 x 7 = 84
12 x 8 = 96
12 x 9 = 108
```

$12 \times 10 = 120$

The screenshot shows the Eclipse IDE interface. At the top, there are tabs for 'ForLoop.java', 'WhileLoop.java', 'DoWhileLoop...', and another tab that is partially visible. Below the tabs is a code editor window containing the following Java code:

```
1 package programming.java.com;
2
3 class MultiplicationTable {
4     public static void main(String[] args) {
5         int n = 12;
6         for (int i = 1; i <= 10; i++) {
7             System.out.println(n + " x " + i + " = " + (n * i));
8         }
9     }
10 }
11
12
```

Below the code editor is a toolbar with icons for 'Problems', 'Javadoc', 'Declaration', 'Console', and 'Eclipse IDE'. The 'Console' tab is selected, showing the output of the program:

```
<terminated> MultiplicationTable [Java Application] D:\eclipse\plugins\org.e
12 x 1 = 12
12 x 2 = 24
12 x 3 = 36
12 x 4 = 48
12 x 5 = 60
12 x 6 = 72
12 x 7 = 84
12 x 8 = 96
12 x 9 = 108
12 x 10 = 120
```

## POST LAB EXERCISE

- ✓ What is the use of if statement?

**The if statement is used to check a condition in a program and control the flow of execution. When the condition is true, the corresponding block of code is executed; otherwise, it is skipped. It helps in decision making and allows programs to behave differently under different conditions.**

- ✓ Difference between if-else and else-if ladder.

**The if-else statement is used when there are only two possible conditions to choose from. The else-if ladder is used to check multiple conditions sequentially. The conditions are evaluated from top to bottom, and the first true condition is executed.**

- ✓ Why is switch statement used?

**The switch statement is used to select one block of code from multiple choices based on the value of a variable. It makes the program more readable and organized when handling many conditions. It is commonly used in menu-driven applications.**

- ✓ Difference between for, while, and do-while loops.

**The for loop is used when the number of iterations is known in advance, while the while loop is used when the number of iterations is not known. The do-while loop executes the loop body at least once because the condition is checked after execution. This makes it different from for and while loops.**

- ✓ Which loop executes at least once?

**The do-while loop executes at least once in a program. This is because the loop body is executed before the condition is evaluated. Even if the condition is false, the loop runs one time.**

### **Result:**

Thus the different control statements were executed successfully with expected output.

### **ASSESSMENT**

Description	Max Marks	Marks Awarded
Pre Lab Exercise	<b>5</b>	
In Lab Exercise	<b>10</b>	
Post Lab Exercise	<b>5</b>	
Viva	<b>10</b>	
<b>Total</b>	<b>30</b>	
<b>Faculty Signature</b>		