

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.

1. Selection (Decision) Statements

Used to make decisions.

- if
- if-else
- else-if
- switch

2. Looping (Iteration) Statements

Used to repeat a block of code.

- for
- while
- do-while

3. Jump Statements

Used to change the normal flow.

- break
- continue
- return

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

Feature	for loop	while loop	do-while loop
Condition check	Before loop starts	Before loop starts	After loop executes
Minimum execution	0 times	0 times	At least 1 time

Feature	for loop	while loop	do-while loop
Best used when	Number of iterations is known	Condition-based looping	Loop must run once

- ✓ What is the use of break and continue?

Break

- Stops the loop immediately
- Control comes out of the loop

continue

- Skips the current iteration
- Loop continues with 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

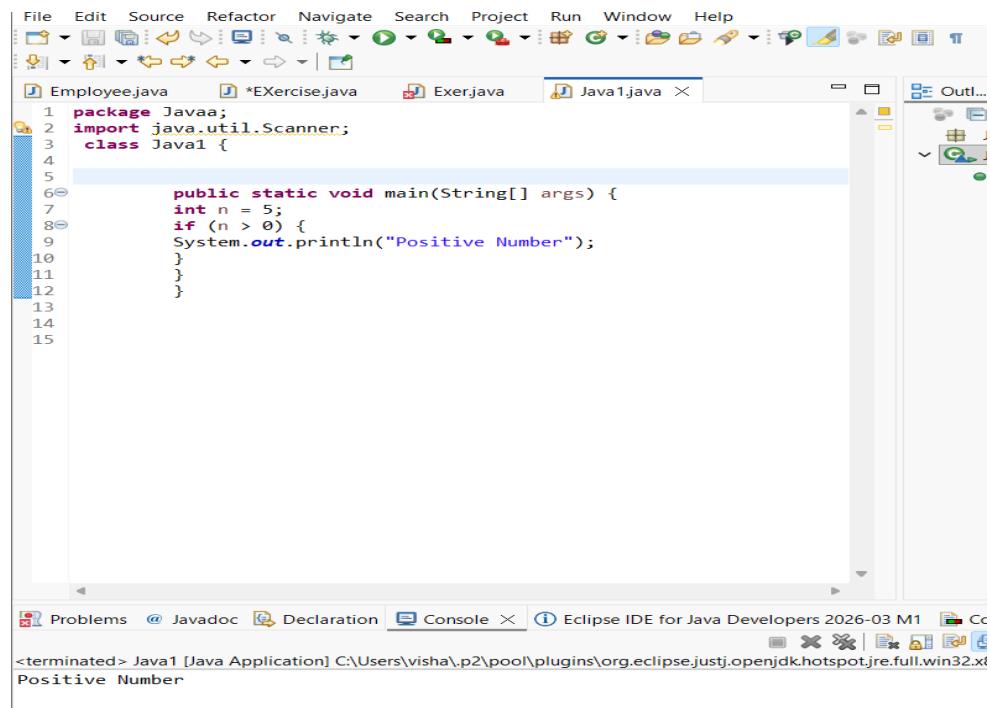
PROGRAMS:

Program 1: Check Whether a Number is Positive

```
class PositiveNumber {  
    public static void main(String[] args) {  
        int n = 5;  
        if (n > 0) {  
            System.out.println("Positive Number");  
        }  
    }  
}
```

Output:

Positive Number



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is a toolbar with various icons. The central workspace displays four tabs: Employee.java, *Exercise.java, Exer.java, and Java1.java. The Java1.java tab is active, showing the following code:

```
1 package Javaa;  
2 import java.util.Scanner;  
3 class Java1 {  
4  
5     public static void main(String[] args) {  
6         int n = 5;  
7         if (n > 0) {  
8             System.out.println("Positive Number");  
9         }  
10    }  
11 }  
12  
13  
14  
15
```

To the right of the workspace is the Out... view, which shows a tree structure with a single entry. At the bottom of the interface, there are tabs for Problems, Javadoc, Declaration, and Console. The Console tab is selected, showing the output: "Positive Number". The status bar at the bottom indicates "Eclipse IDE for Java Developers 2026-03 M1" and the path "C:\Users\visha\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64".

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 is a tab bar with several tabs: Employee.java, *ExExcise.java, Exerjava, and Java1.java (which is currently selected). Below the tabs is a code editor window containing Java code. The code defines a class Java1 with a main method that prints "Even Number" if the input is even, and "Odd Number" if it is odd. Lines 7 through 14 are highlighted in blue, indicating the code being run. The code editor has line numbers on the left. At the bottom of the screen is a toolbar with icons for Problems, Javadoc, Declaration, Console, and Eclipse IDE for Java Developers. The Console tab is selected, showing the output: <terminated> Java1 [Java Application] C:\Users\visha\.p2\pool\plugins\org.eclipse.justj.open. The output itself is "Even Number".

```

1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
4
5
6
7     public static void main(String[] args) {
8         int n = 6;
9         if (n % 2 == 0)
10             System.out.println("Even Number");
11         else
12             System.out.println("Odd Number");
13     }
14 }
15
16
17

```

Program 3: Find Largest of Two Numbers

```

class LargestTwo {
    public static void main(String[] args) {

```

```

int a = 10, b = 20;
if (a > b)
    System.out.println("A is largest");
else
    System.out.println("B is largest");
}
}

```

Output:

B is largest

The screenshot shows the Eclipse IDE interface. At the top, there are tabs for "Employee.java", "*EXercise.java", "Exer.java", and "Java1.java". The "Java1.java" tab is active, displaying the following Java code:

```

1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
4
5
6
7     public static void main(String[] args) {
8         int a = 10, b = 20;
9         if (a > b)
10             System.out.println("A is largest");
11         else
12             System.out.println("B is largest");
13     }
14 }

```

Below the code editor is the "Console" view, which shows the output of the program:

```

<terminated> Java1 [Java Application] C:\Users\visha\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jdk11\bin> B is largest

```

Program 4: Grade Calculation

```
class Grade {
```

```

public static void main(String[] args) {
    int marks = 75;
    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 B

The screenshot shows the Eclipse IDE interface. At the top, there are tabs for Employee.java, *Exercise.java, Exer.java, and Java1.java (which is currently selected). Below the tabs is a code editor window displaying the Java code provided above. The code is syntax-highlighted, with keywords like package, import, public, static, void, if, else, and System.out.println in different colors. The code itself is a simple grade determination program. At the bottom of the screen, there is a toolbar with icons for Problems, Javadoc, Declaration, Console, and Eclipse IDE for Java. The Console tab is active, showing the output of the program: '<terminated> Java1 [Java Application] C:\Users\visha\p2\pool\plugins\org.eclipse.just Grade B'. This indicates that the program ran successfully and printed 'Grade B' to the console.

```

1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
4     public static void main(String[] args) {
5         int marks = 75;
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 }
16

```

Program 5: Day of the Week

```
class DaySwitch {
```

```

public static void main(String[] args) {
    int day = 3;
    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;
        default: System.out.println("Invalid Day");
    }
}
}
}

```

Output:

Wednesday

The screenshot shows the Eclipse IDE interface. At the top, there are tabs for 'Employee.java', '*Exercise.java', 'Exer.java', and 'Java1.java'. The 'Java1.java' tab is active, displaying the Java code provided in the previous block. The code defines a class 'Java1' with a main method that prints 'Wednesday' to the console. Below the editor, the 'Console' view is open, showing the output: '<terminated> Java1 [Java Application] C:\Users\visha\p2\pool\plugins\org.eclipse.just Wednesday'. The bottom of the screen shows the Eclipse toolbar with icons for Problems, Javadoc, Declaration, Console, and Help.

```

1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
4     public static void main(String[] args) {
5         int day = 3;
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            default: System.out.println("Invalid Day");
13        }
14    }
15 }
16

```

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 with the following details:

- Project Explorer:** Shows files Employee.java, *Exercise.java, Exer.java, and Java1.java.
- Java1.java Content:** The code for the ForLoop class is displayed in the editor.
- Outline View:** Shows the Java1 class and its main() method.
- Console Output:** The output of the program is shown in the console, displaying the numbers 1 through 5.
- Status Bar:** Shows "terminated > Java1 [Java Application] C:\Users\visha.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.9.v".

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. At the top, there are tabs for Employee.java, *Exercise.java, Exer.java, and Java1. The Java1 tab is active, displaying the following Java code:

```

1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
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
13

```

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

```

terminated> Java1 [Java Application] C:\Users\visha\.p2\pool\plugins\orc
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. At the top, there are tabs for Employee.java, *Exercise.java, Exer.java, and Java1.java. The Java1.java tab is active, displaying the following code:

```

1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
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
13

```

Below the editor, the Eclipse toolbar includes icons for Problems, Javadoc, Declaration, and Console. The Console tab is selected, showing the output of the program:

```

<terminated> Java1 [Java Application] C:\Users\visha\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
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 tabs for Employee.java, *Exercise.java, Exer.java, and Java1.java. The Java1.java tab is active, displaying the following Java code:

```

1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
4
5     public static void main(String[] args) {
6         int sum = 0;
7         for (int i = 1; i <= 5; i++) {
8             sum = sum + i;
9         }
10        System.out.println("Sum = " + sum);
11    }
12 }
13
14

```

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

```

<terminated> Java1 [Java Application] C:\Users\visha\p2\pool\plugins\org.eclipse.justj.or
Sum = 15

```

Program 10: Multiplication Table of a Number

```

class MultiplicationTable {
    public static void main(String[] args) {
        int n = 5;
        for (int i = 1; i <= 10; i++) {
            System.out.println(n + " x " + i + " = " + (n * i));
        }
    }
}

```

```
}
```

```
}
```

```
}
```

Output:

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

The screenshot shows the Eclipse IDE interface. At the top, there are tabs for "Employee.java", "*Exercise.java", "Exer.java", and "Java1.java". The "Java1.java" tab is active, displaying the following Java code:

```
1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
4     public static void main(String[] args) {
5         int n = 5;
6         for (int i = 1; i <= 10; i++) {
7             System.out.println(n + " x " + i + " = " + (n * i));
8         }
9     }
10 }
11
12
13
```

Below the code editor, the "Console" view is open, showing the output of the program:

```
<terminated> Java1 [Java Application] C:\Users\visha\p2\pool\plugins\org.eclipse.justj.openjdk
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 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

```

POST LAB EXERCISE

- ✓ What is the use of if statement?
 - The `if` statement is used to check a condition.
 - If the condition is true, the code inside `if` is executed.
 - If it is false, the code is skipped.
- ✓ Difference between if-else and else-if ladder.

Feature	if-else	else-if ladder
Conditions	Only one condition	Multiple conditions
Choices	Two choices	Many choices
Execution	Either if or else	First true condition executes

- ✓ Why is switch statement used?
 - `switch` is used to select one option from many choices.
 - It is easier and cleaner than multiple else-if statements.
 - Works with values like int, char, String.
- ✓ Difference between for, while, and do-while loops.

Feature	for	while	do-while
Condition check	Before loop	Before loop	After loop

Feature	for	while	do-while
Executes	0 or more times	0 or more times	At least once
Best used when	Count is known	Condition-based	Must run once

- ✓ Which loop executes at least once?

The do-while loop executes at least once.

Because the condition is checked after the loop body.

Result:

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

ASSESSMENT

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