

## 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

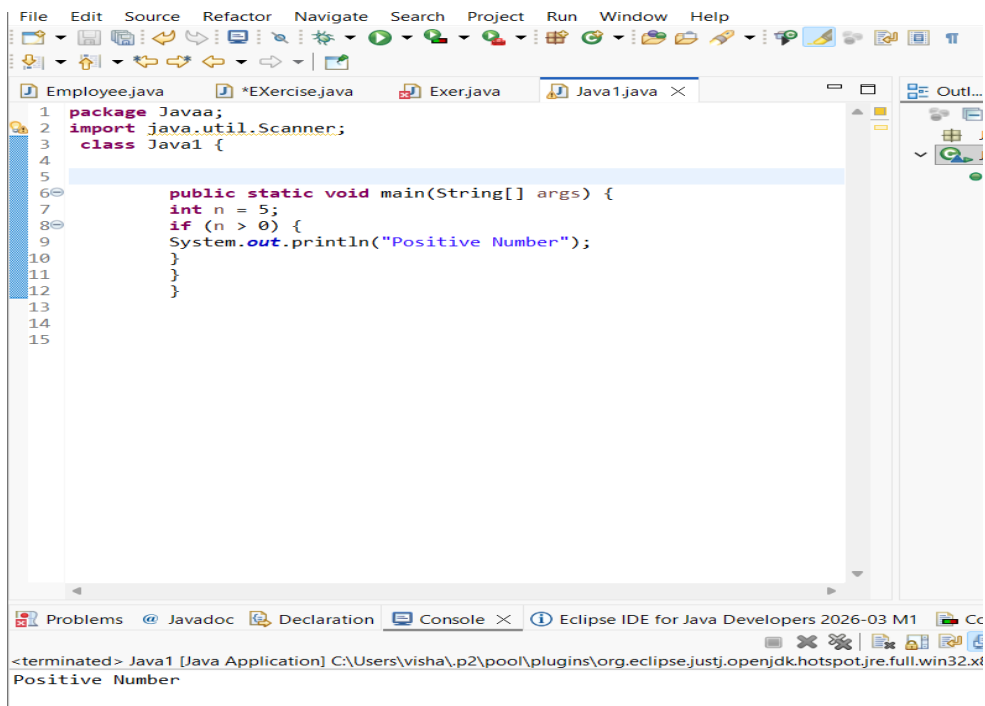
## PROGRAMS:

### 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



### 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");

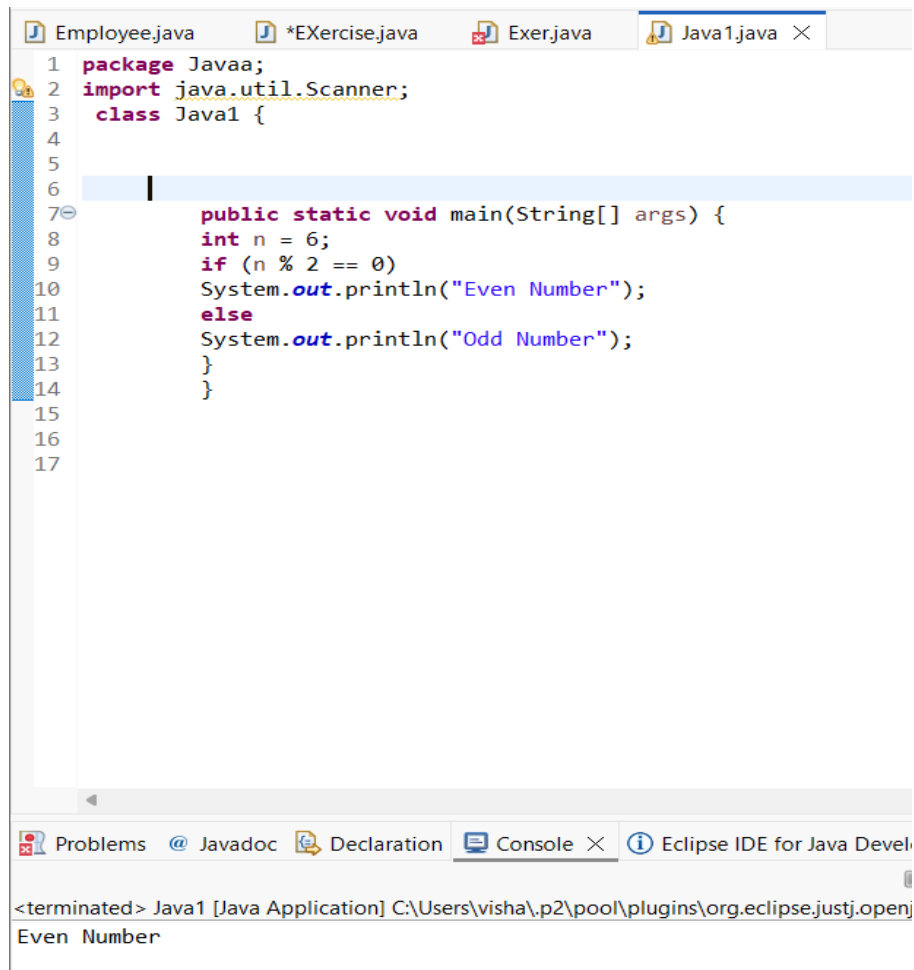
}

}

```

### Output:

Even Number



```

Employee.java *EXercise.java Exer.java Java1.java ×
1 package Javaa;
2 import java.util.Scanner;
3 class Java1 {
4
5
6
7
8     public static void main(String[] args) {
9         int n = 6;
10        if (n % 2 == 0)
11            System.out.println("Even Number");
12        else
13            System.out.println("Odd Number");
14        }
15    }
16
17
Problems Javadoc Declaration Console × Eclipse IDE for Java Devel
<terminated> Java1 [Java Application] C:\Users\visha\p2\pool\plugins\org.eclipse.justj.openj
Even Number

```

### 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");

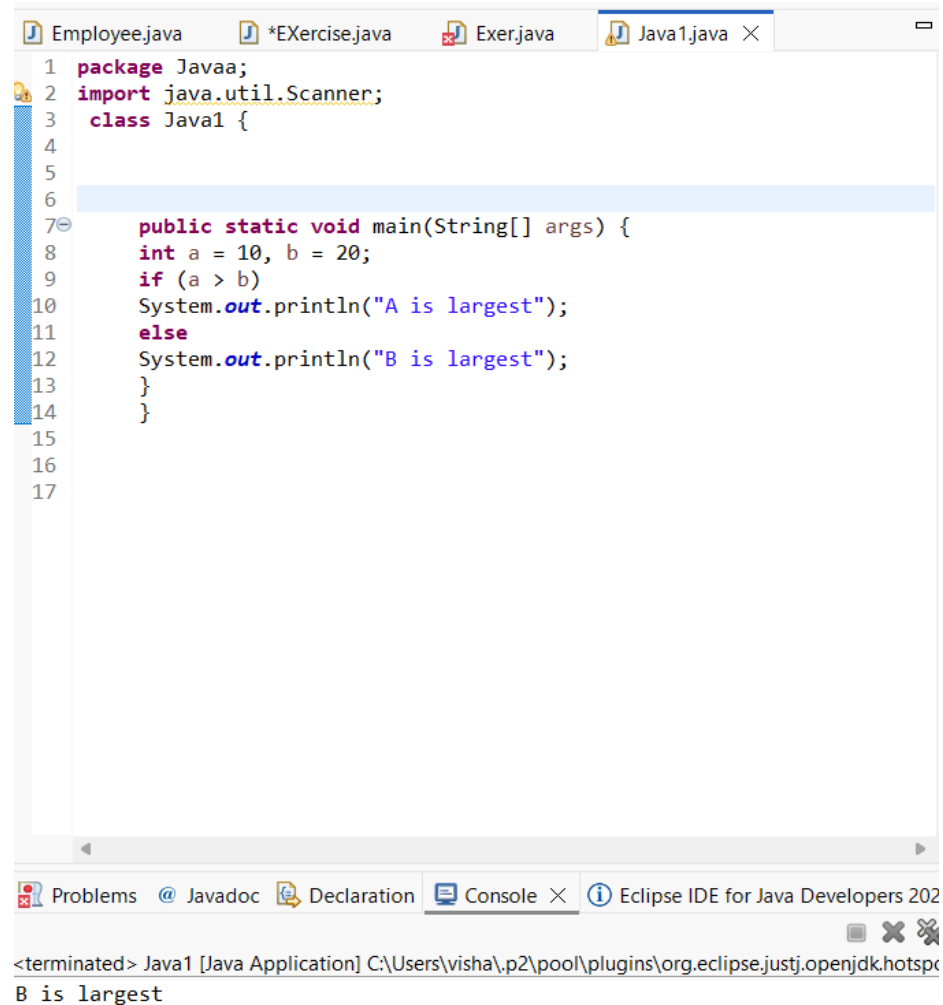
}

}

```

### Output:

B is largest



The screenshot shows the Eclipse IDE with a project named 'Java1'. The main editor displays 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    }
15
16
17

```

The bottom of the IDE shows the 'Console' tab with the output:

```

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

```

### Program 4: Grade Calculation

```

class Grade {

```

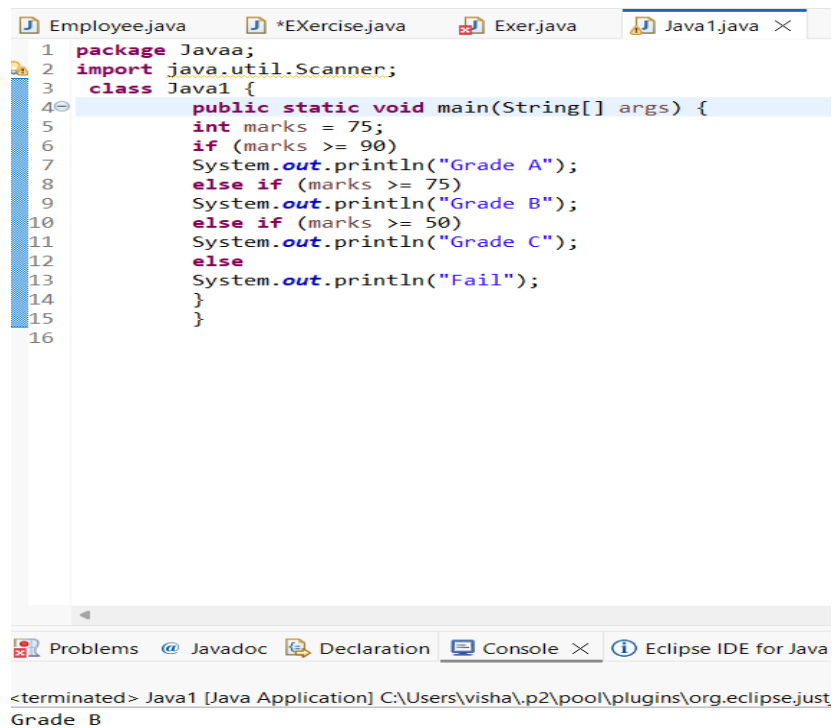
```

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 with a Java project. The main editor displays the following code:

```

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

```

The bottom of the IDE shows the Console view with the output:

```

<terminated> Java1 [Java Application] C:\Users\visha\.p2\pool\plugins\org.eclipse.just
Grade B

```

### 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 with a Java project. The editor displays the following code in `Java1.java`:

```

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

```

The bottom of the IDE shows the Console view with the output:

```

<terminated> Java1 [Java Application] C:\Users\visha\.p2\pool\plugins\org.eclipse.just
Wednesday

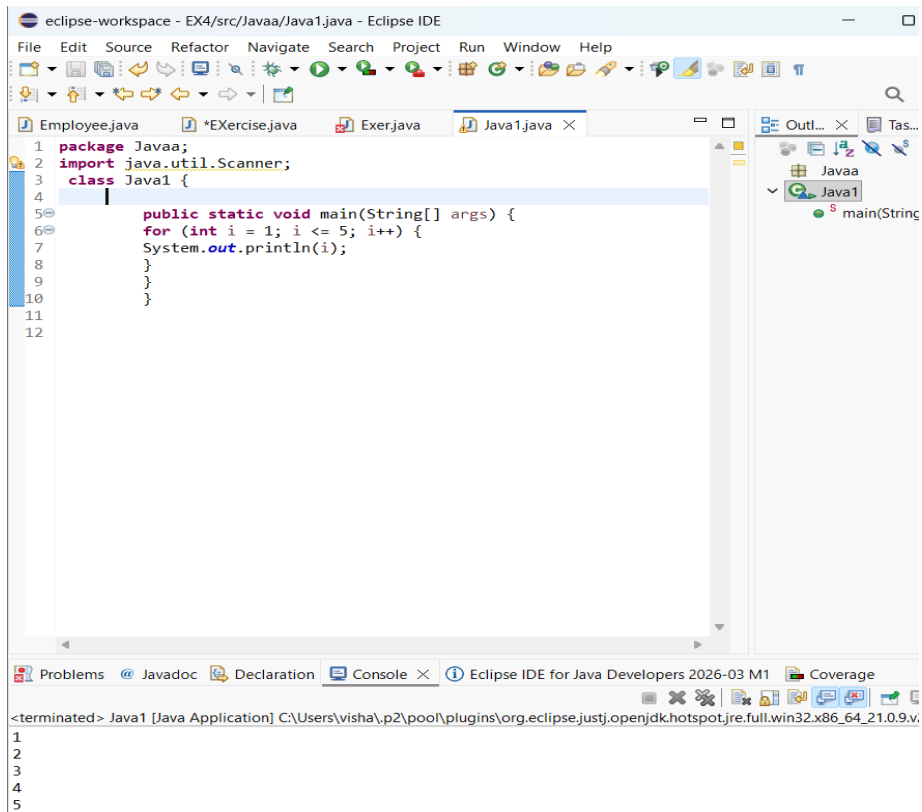
```

### 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



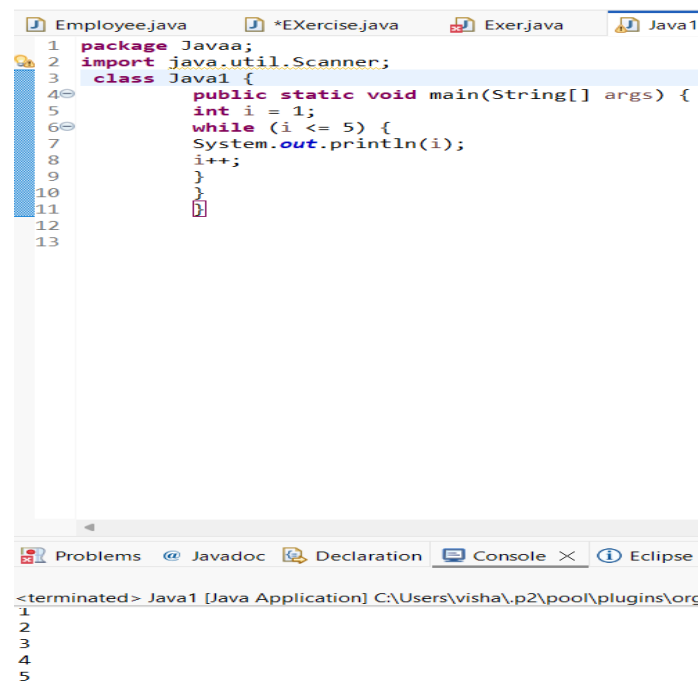
### Program 7: Print Numbers from 1 to 5



```
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 with a Java project. The editor displays 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         while (i <= 5) {  
7             System.out.println(i);  
8             i++;  
9         }  
10    }  
11 }  
12  
13
```

The bottom of the IDE shows the 'Console' tab with the output:

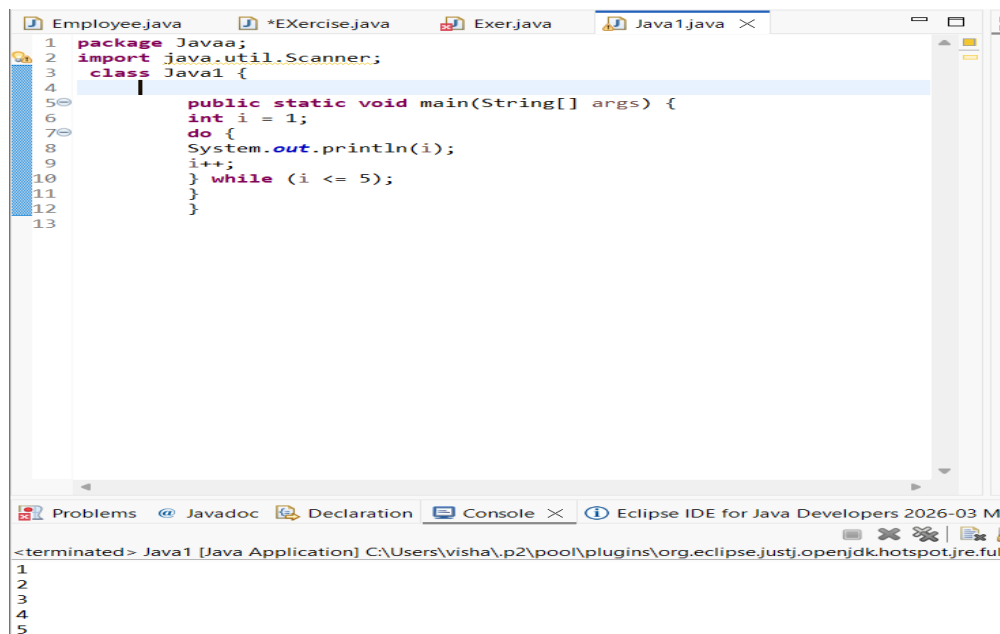
```
<terminated> Java1 [Java Application] C:\Users\visha\.p2\pool\plugins\org  
1  
2  
3  
4  
5
```

### 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 with a Java project. The main editor displays the following code in `Java1.java`:

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

The bottom of the IDE shows the `Console` view with the output of the program:

```
<terminated> Java1 [Java Application] C:\Users\visha\.p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full\jre\bin\java.exe  
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);

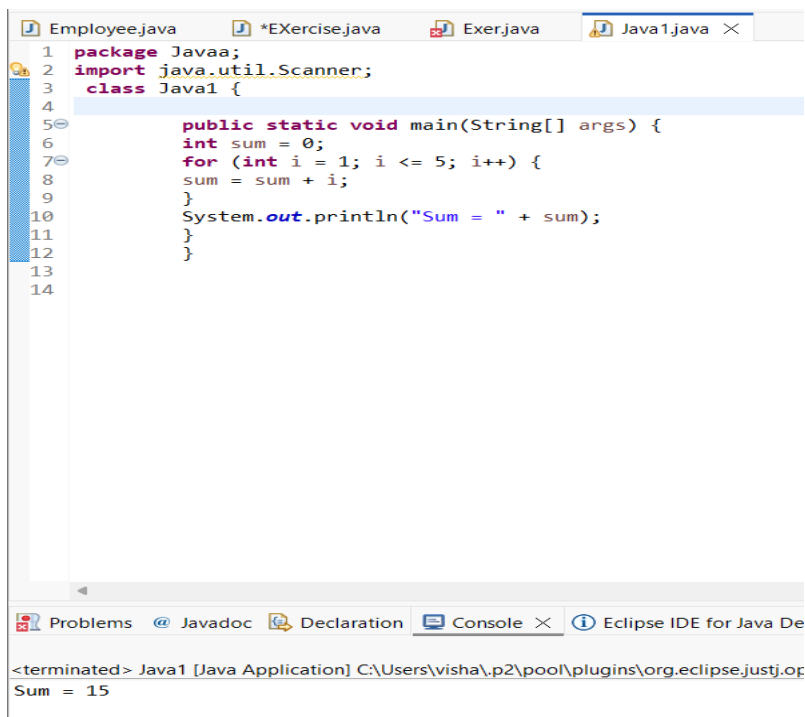
}

}

```

### Output:

Sum = 15



The screenshot shows the Eclipse IDE with a Java project. The editor displays a file named `Java1.java` with the following 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

```

The IDE's console at the bottom shows the output of the program:

```

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

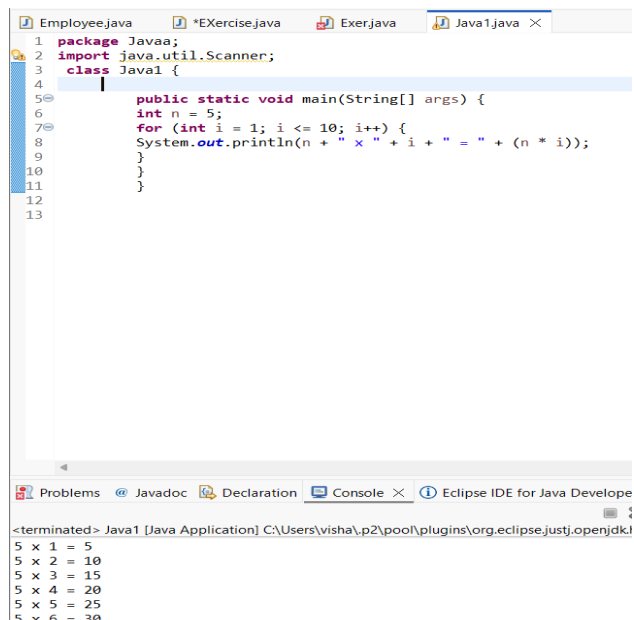
5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50



The screenshot shows the Eclipse IDE with a Java project. The editor displays a file named `Java1.java` with the following code:

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

The bottom of the IDE shows the `Console` tab with the following output:

```
<terminated> Java1 [Java Application] C:\Users\visha\p2\pool\plugins\org.eclipse.justi.openjdk.l  
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

<b>Feature</b>	<b>for</b>	<b>while</b>	<b>do-while</b>
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**

<b>Description</b>	<b>Max Marks</b>	<b>Marks Awarded</b>
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>		