

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.

Selection statements

- if
- if-else
- switch

Iteration (looping) statements

- for
- while
- do-while

Jump statements

- break
- continue
- return

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

Loop	Condition Check	Execution
for	Before loop starts	Used when number of iterations is known
while	Before loop starts	Used when iterations are not fixed
do-while	After loop executes	Executes at least once

What is the use of break and continue?

break: Stops the loop or switch statement **completely**.

continue: Skips the current iteration and moves to the **next iteration** of the loop.

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

```

ReadName.java 1 ×
1  class PositiveNumber {      ReadName.java is a non-project file, only syntax errors are reported
2  |  Run | Debug
3  |  public static void main(String[] args) {
4  |  |  int n = 5;
5  |  |  if (n > 0) {
6  |  |  |  System.out.println("Positive Number");
7  |  |  }
8  |  }
9

PROBLEMS 1 OUTPUT DEBUG CONSOLE EXPLORER TERMINAL PORTS

PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:60892' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscodews_7b297\jdt_ws\jdt.ls-java-project\bin' 'PositiveNumber'
Positive Number
PS C:\Users\ragur>

```

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 VS Code interface with the following details:

- Editor:** The file "ReadName.java 1" is open, containing the following Java code:

```
1  class Evenodd {    ReadName.java is a non-project file, only syntax errors are reported
2      public static void main(String[] args) {
3          int n = 6;
4          if (n % 2 == 0)
5              System.out.println(x: "Even Number");
6          else
7              System.out.println(x: "Odd Number");
8      }
9  }
```
- Terminal:** The terminal tab is active, showing the command line and its output:

```
PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '--agentlib:jdwpt=transport=dt_socket,server=n,suspend=y,address=localhost:60899' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscodews_7b297\jdt_ws\jdt.ls-java-project\bin' 'EvenOdd'
Even Number
PS C:\Users\ragur>
```

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 Visual Studio Code interface. The main area displays a Java file named `LargestTwo.java` with the following code:

```
1 class LargestTwo {    ReadName.java 1
2     public static void main(String[] args) {
3         int a = 10, b = 20;
4         if (a > b)
5             System.out.println("A is largest");
6         else
7             System.out.println("B is largest");
8     }
9 }
```

Below the code editor, the terminal window shows the following output:

```
PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:54827' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscodews_7b297\jdt_ws\jdt.ls-java-project\bin' 'LargestTwo'
B is largest
PS C:\Users\ragur>
```

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 a Java code editor with the file 'ReadName.java' open. The code defines a class 'Grade' with a main method that prints the grade based on marks. The code is color-coded, with keywords in blue, comments in green, and strings in red. Below the editor is a terminal window showing the output of running the program with input '75'. The terminal tabs at the bottom include PROBLEMS, OUTPUT, DEBUG CONSOLE, EXPLORER, TERMINAL (which is selected), and PORTS.

```
1 class Grade {    ReadName.java is a non-project file, only syntax errors are reported
2     Run | Debug
3     public static void main(String[] args) {
4         int marks = 75;
5         if (marks >= 90)
6             System.out.println("Grade A");
7         else if (marks >= 75)
8             System.out.println("Grade B");
9         else if (marks >= 50)
10            System.out.println("Grade C");
11        else
12            System.out.println("Fail");
13    }
14 }
```

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE EXPLORER TERMINAL PORTS

PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '--agentlib:jdwp=transport=dt_socket,server=n,suspend=y,ac
e-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscodews_7b297\jdt_ws\jdt.ls\jav
Grade B
PS C:\Users\ragur>
```

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 a Java code editor with a dark theme. A file named 'ReadName.java' is open, containing the following code:

```
1 class Dayswitch {    ReadName.java is a non-project file, only syntax errors are reported
2     Run | Debug
3     public static void main(String[] args) {
4         int day = 3;
5         switch (day) {
6             case 1:
7                 System.out.println(x: "Monday");
8                 break;
9             case 2:
10                 System.out.println(x: "Tuesday");
11                 break;
12             case 3:
13                 System.out.println(x: "Wednesday");
14                 break;
15             case 4:
16                 System.out.println(x: "Thursday");
17                 break;
18             case 5:
19                 System.out.println(x: "Friday");
20                 break;
21         default:
22     }
23 }
```

Below the code editor, there is a terminal window showing the command run and its output:

```
PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:557
e-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscodews_7b297\jdt_ws\jdt.ls-java-project\bin' 'Day
Wednesday
PS C:\Users\ragur>
```

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 Visual Studio Code interface. In the top editor pane, there is a Java file named `ReadName.java` with the following content:

```
1 < class ForLoop {    ReadName.java is a non-project file, only syntax errors are reported
2 <     Run | Debug
3 <     public static void main(String[] args) {
4 <         for (int i = 1; i <= 5; i++) {
5 <             System.out.println(i);
6 <         }
7 >     }
8 > }
```

The bottom pane shows the terminal output:

```
PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '-agentlib:jdwpt=transport=dt_socket,serv=e-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscodews_7b297\1
2
3
4
5
PS C:\Users\ragur>
```

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 VS Code interface with the following details:

- Editor:** A Java file named "ReadName.java" is open. The code contains a class named "ForLoop" with a main method that prints numbers from 1 to 5.
- Terminal:** The terminal window shows the command used to run the Java application and the resulting output. The output is:

```
PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,sus-e-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscodews_7b297\jdt_ws\j
1
2
3
4
5
PS C:\Users\ragur>
```
- Bottom Bar:** The bottom bar shows the navigation tabs: PROBLEMS (1), OUTPUT, DEBUG CONSOLE, EXPLORER, TERMINAL (selected), and PORTS.

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 a Java code editor with a file named 'ReadName.java'. The code contains a 'DoWhileLoop' class with a main method that prints integers from 1 to 5. Below the editor is a terminal window showing the execution of the program and its output.

```
1 class DoWhileLoop {    ReadName.java is a non-project file, only syntax errors are reported  
2     public static void main(String[] args) {  
3         int i = 1;  
4         do {  
5             System.out.println(i);  
6             i++;  
7         } while (i <= 5);  
8     }  
9 }  
10  
PROBLEMS 1 OUTPUT DEBUG CONSOLE EXPLORER TERMINAL PORTS  
PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,ad  
e-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscodews_7b297\jdt_ws\jdt.ls\jav  
1  
2  
3  
4  
5  
PS C:\Users\ragur>
```

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 Visual Studio Code interface. At the top, there's a tab bar with 'ReadName.java 1 X'. Below it is the code editor containing the following Java code:

```
1 class SumNumbers {    ReadName.java is a non-project file, only syntax errors are reported
2     Run | Debug
3     public static void main(String[] args) {
4         int sum = 0;
5         for (int i = 1; i <= 5; i++) {
6             sum = sum + i;
7         }
8         System.out.println("Sum = " + sum);
9     }
10 }
```

Below the code editor is a terminal window showing the output of the program:

```
PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,address=4444,server=y,suspend=n' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\Local\Temp\vscod
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:

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$5 \times 10 = 50$$

The screenshot shows a Java code editor with a dark theme. A file named `ReadName.java` is open, containing the following code:

```
1  class MultiplicationTable {    ReadName.java is a non-project file, only
2      public static void main(String[] args) {
5          |          |    System.out.println(n + " x " + i + " = " + (n * i));
6          |
7      }
8  }
9
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS (1), OUTPUT, DEBUG CONSOLE, EXPLORER, TERMINAL (underlined), and PORTS.

The TERMINAL tab displays the following command-line output:

```
PS C:\Users\ragur> & 'C:\Program Files\Java\jdk-25.0.2\bin\java.exe' '-agentlib:jd
e-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ragur\AppData\
able'
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
PS C:\Users\ragur>
```

POST LAB EXERCISE

- ✓ What is the use of if statement?

It is used to check a condition and execute a block of code.

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

- If-else checks only one condition and execute the block meanwhile the if-else ladder check multiple if conditions and execute one if block and skips the rest.

- ✓ Why is switch statement used?

- It is used when we have to select one from many choices.
- It is easier and cleaner than if else.

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

Loop	Condition Check	Usage
for	Before loop starts	When number of iterations is known
while	Before loop starts	When iterations are not known
do-while	After loop runs	When loop must run at least once

- ✓ Which loop executes at least once?

- Do-while loop executes at least once because it checks condition after the loop enters the 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		