

## Do-While Loop in Java

### Detailed Notes

The do-while loop is a control flow statement that executes a block of code at least once and then repeats it while a boolean condition remains true. The key difference from a while loop is that the condition is evaluated after the execution of the loop body.

### Syntax

```
java
do {
    // statements to execute
} while (condition);
```

The loop body runs first, regardless of condition.

After each execution, the condition is checked.

If condition is true, the loop continues; otherwise, it exits.

### Execution Flow:

Execute the block.

Evaluate condition.

If true, repeat block.

If false, exit loop.

### Example:

```
java
int count = 1;
do {
    System.out.println("Count is: " + count);
    count++;
} while (count <= 5);
```

Output:

```
text
Count is: 1
Count is: 2
Count is: 3
Count is: 4
Count is: 5
```

### Characteristics

Guarantees at least one execution even if the condition is false initially.

Useful when you want to prompt users at least once before checking conditions.

Often used in menu-driven programs or interactive applications.

### Sample Programs

Simple do-while example:

```
java
int i = 0;
do {
    System.out.println(i);
    i++;
} while (i < 3);
```

User input validation example:

```
java
Scanner scanner = new Scanner(System.in);
int number;
do {
```

```
        System.out.print("Enter a positive number: ");
        number = scanner.nextInt();
    } while (number <= 0);
    System.out.println("You entered " + number);
```

Interview Questions and Answers: Do-While Loop

What is a do-while loop, and how does it differ from a while loop?

A do-while loop executes the code block once before checking the condition, ensuring at least one execution. A while loop, conversely, checks the condition before the first execution and may not run the code at all if the condition is initially false.

When is the do-while loop preferred in real-world applications?

It is ideal when you want the loop body to execute at least once, such as in user input validation, where you display a prompt, take input, and only then check if it meets a condition.

Can a do-while loop become infinite? How?

Yes, if the condition always evaluates to true or if no statements inside the loop update variables affecting the condition, the loop will run forever causing an infinite loop.

How does variable scope behave inside a do-while loop?

Variables declared inside the do-while loop are scoped to the block. Variables declared outside the loop can be accessed and modified inside the loop.

Is it syntactically valid for the do-while loop to have an empty body?

It is possible but rarely useful. For example, `do ; while(condition);` runs the loop while condition holds, doing nothing inside.