While Loop in Java

SWIPE ----->

comet_11
@payalmanghnani

While Loop :-

The while loop is a control flow statement that repeats a block of code as long as a given condition is true. It is a versatile statement that can be used to implement a variety of looping tasks.

GROUP

Syntax:-

```
while (condition) {
    // block of code to be executed repeatedly
}
```

▶ The condition is a boolean expression that is evaluated before each iteration of the loop. If the condition is true, the block of code is executed. If the condition is false, the loop terminates.

GROUP

Example

The following code prints the numbers from 1 to 10 using a while loop:

```
int i = 1;
while (i <= 10) {
  System.out.println(i);
  i++;
}
```

Heading 2

GROUP

Example Explantation

This code will first initialize the variable i to 1. Then, the loop will start. The condition i <= 10 will be evaluated. If the condition is true, the block of code will be executed. The block of code will print the current value of i. Then, the value of i will be incremented by 1. The loop will then repeat. The process will continue until the condition i <= 10 is false. When the condition is false, the loop will terminate.

Flow of Execution : -
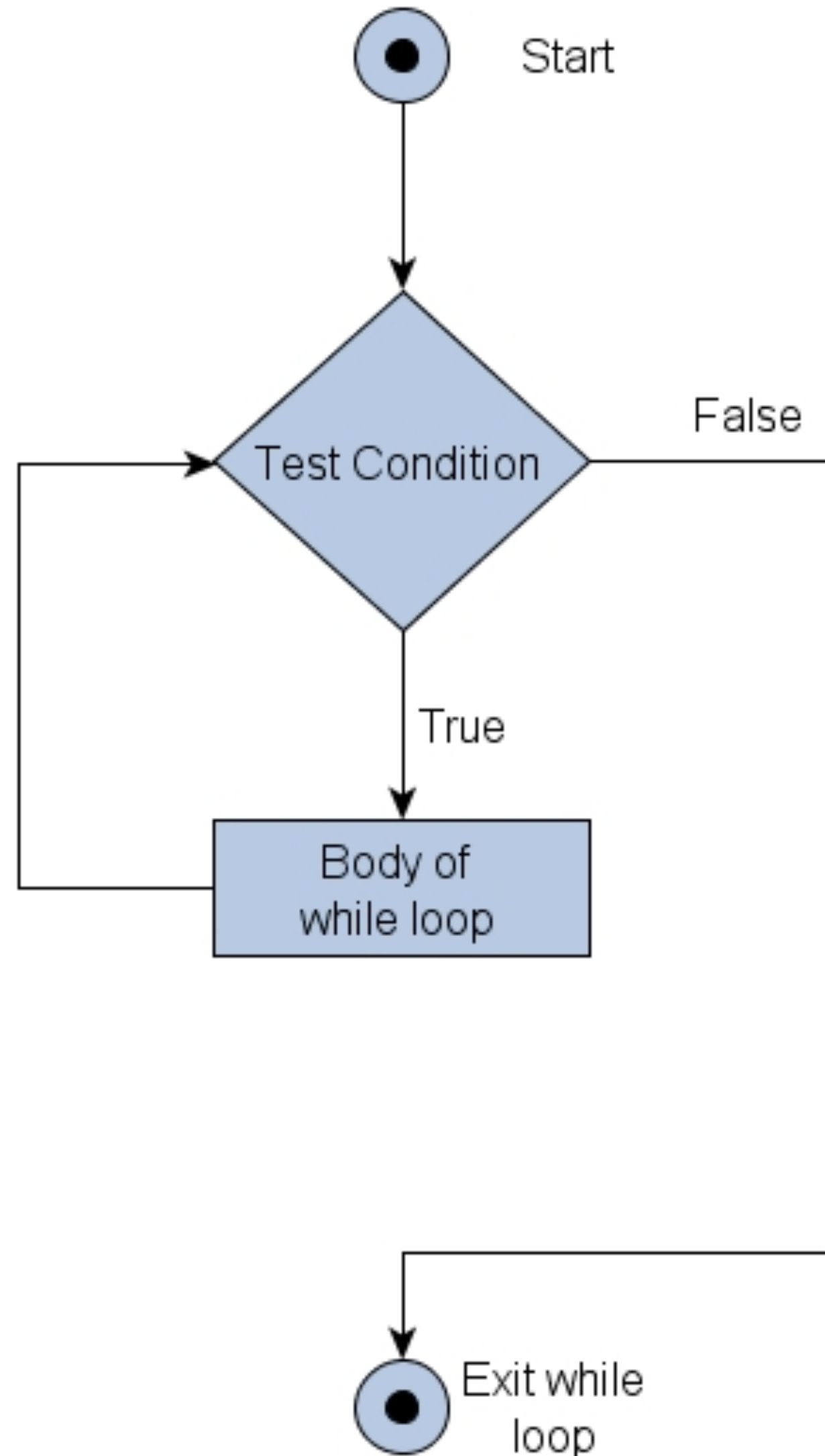
1. The condition is evaluated.

2. If the condition is true, the block of code is executed.

3. After the block of code is executed, the condition is evaluated again.

4. Steps 2 and 3 are repeated until the condition is false.

Termination

The while loop terminates when the condition becomes false. This means that the block of code will not be executed again.

Advantages of Using While Loops :-

- Continuous Iteration
- Flexibility in Loop Conditions
- Handling Dynamic Scenarios
- Simplifying Code Structure
- Avoiding Code Duplication
- User Input Processing
- Efficient Resource Utilization

Disadvantages:-

Potential for Infinite Loops: One of the main disadvantages of while loops is the risk of creating infinite loops. If the loop's exit condition is not properly defined or updated within the loop, the program can get stuck in an endless loop, leading to a crash or unresponsive application.

Initialization Complexity: Unlike for loops, which allow you to initialize variables within the loop declaration, while loops require you to initialize variables outside the loop. This can lead to code duplication or confusion if the initialization logic is not straightforward.