Execution Flow and Flow Control Statements

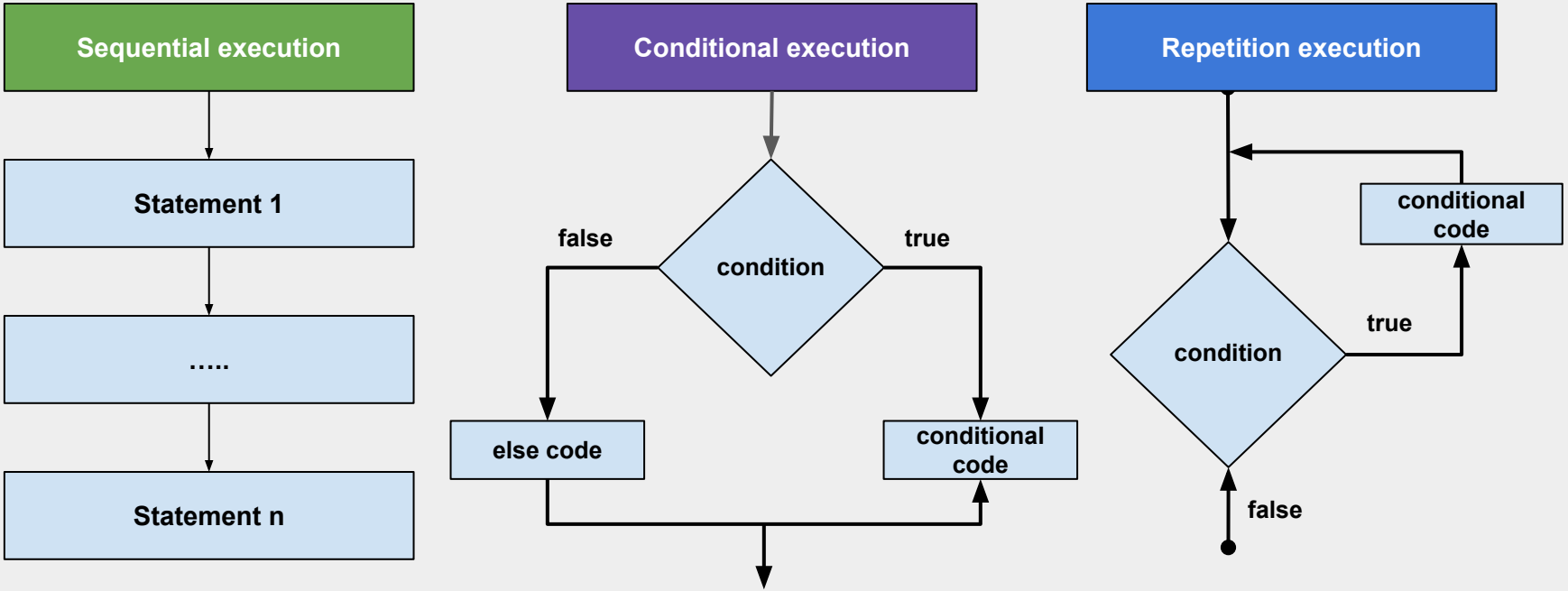
Outline

1. Execution flow
2. Flow Control Statements
3. If, If else and nested if
4. Switch case
5. Loops
    - do while
    - while loop
    - for loop
    - Break, continue
    - Infinite loops
6. Project
7. Interview Questions
8. Summary

Execution Flow

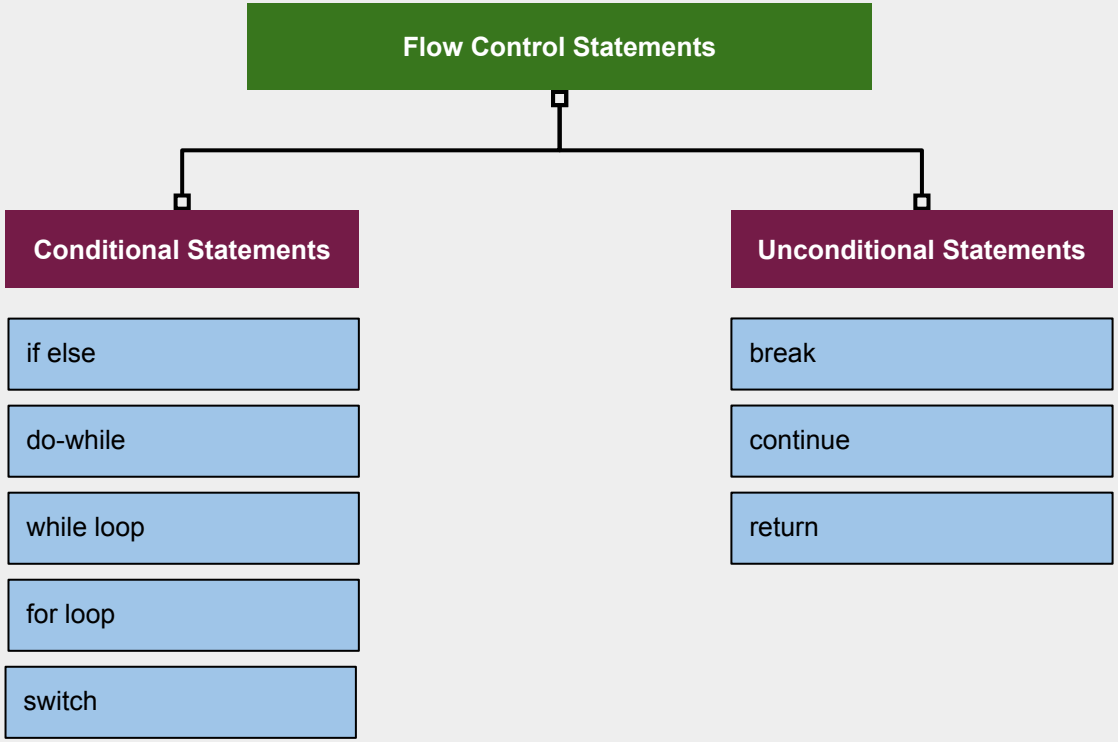
In a given block, the program is executed sequentially one statement at a time starting from the first statement at the top and proceeding toward the bottom.

There are **three type of execution flows** in java

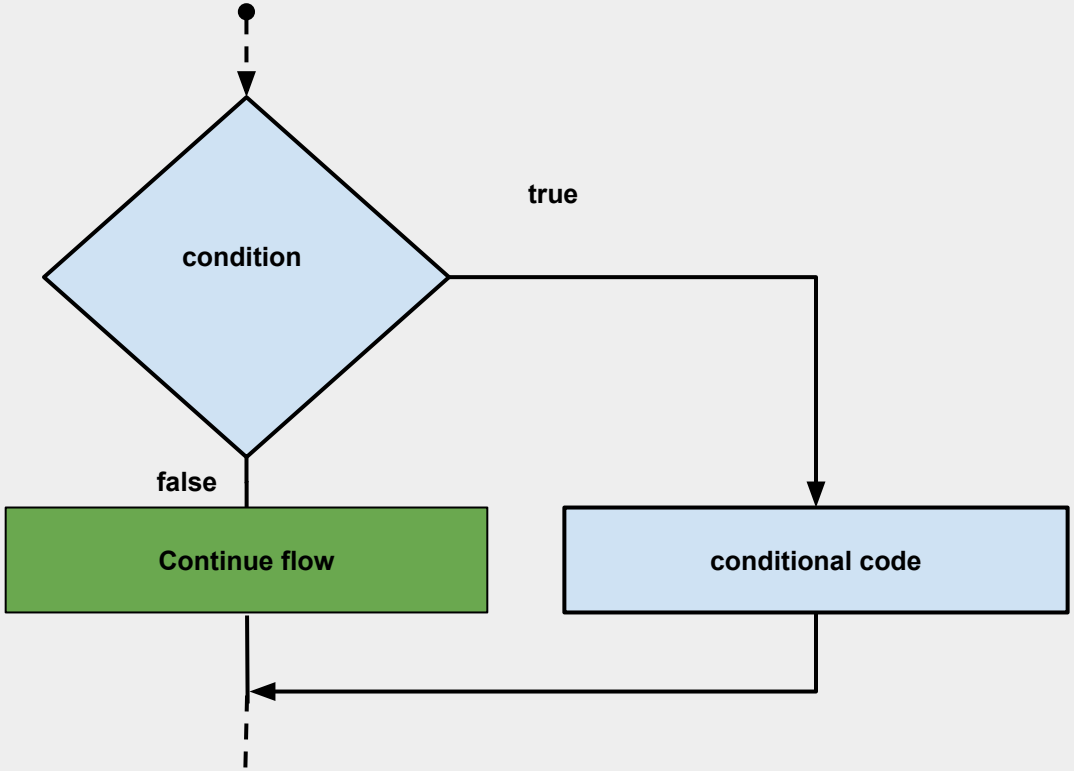


Flow Control Statements

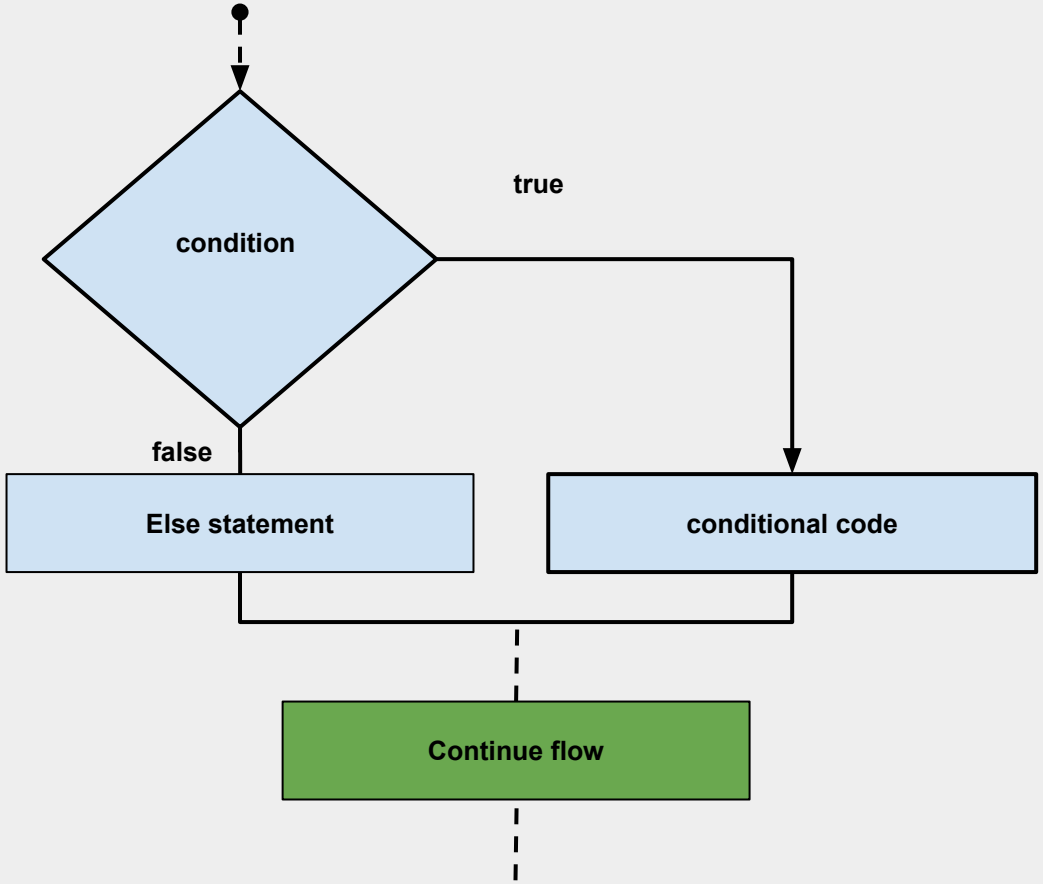
Flow control statements in Java are those statements that change the flow of execution and provide better control to the programmer on the flow of execution in the program. Control statements in java programming are used to write better and complex programs.



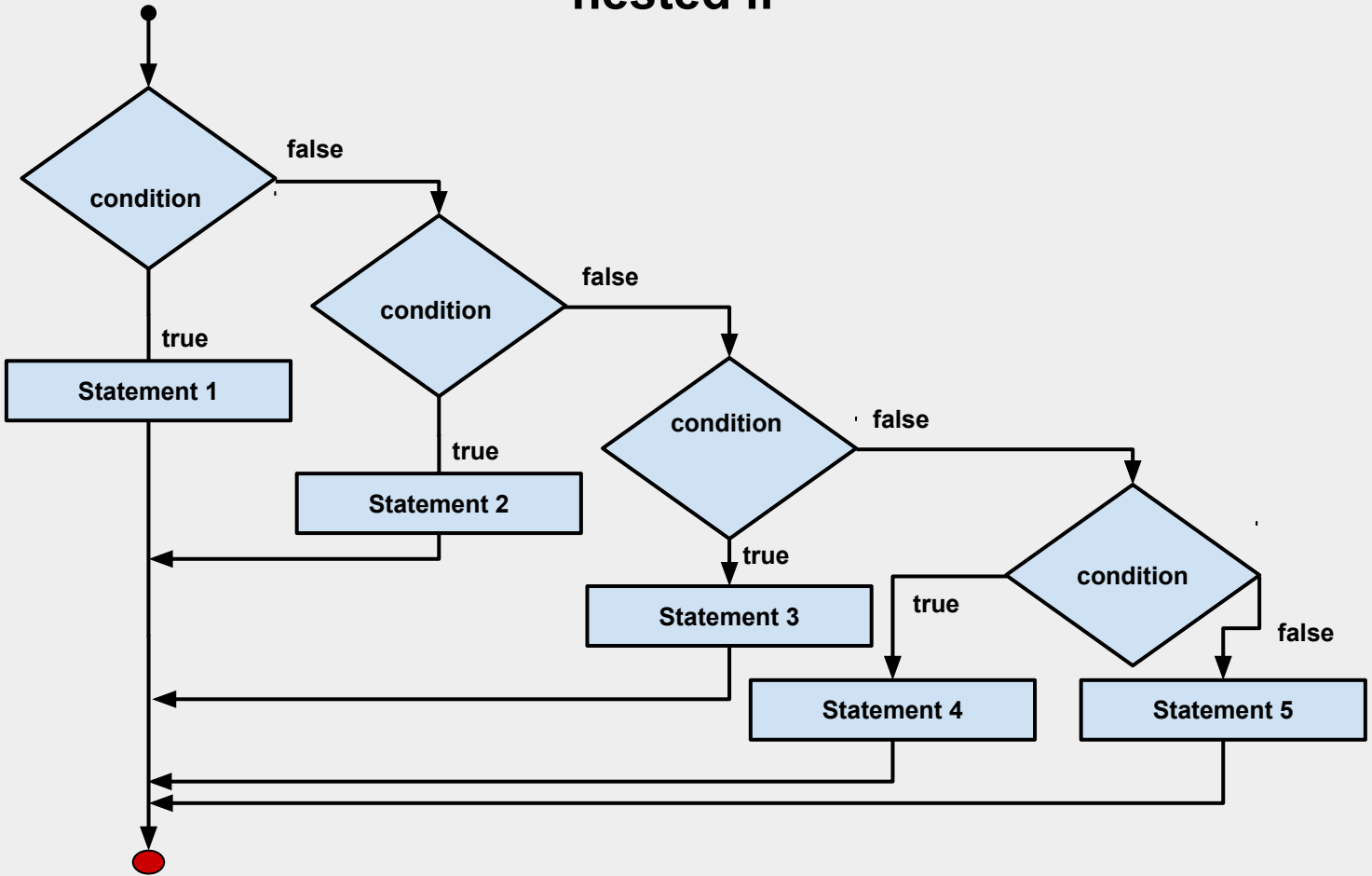
if



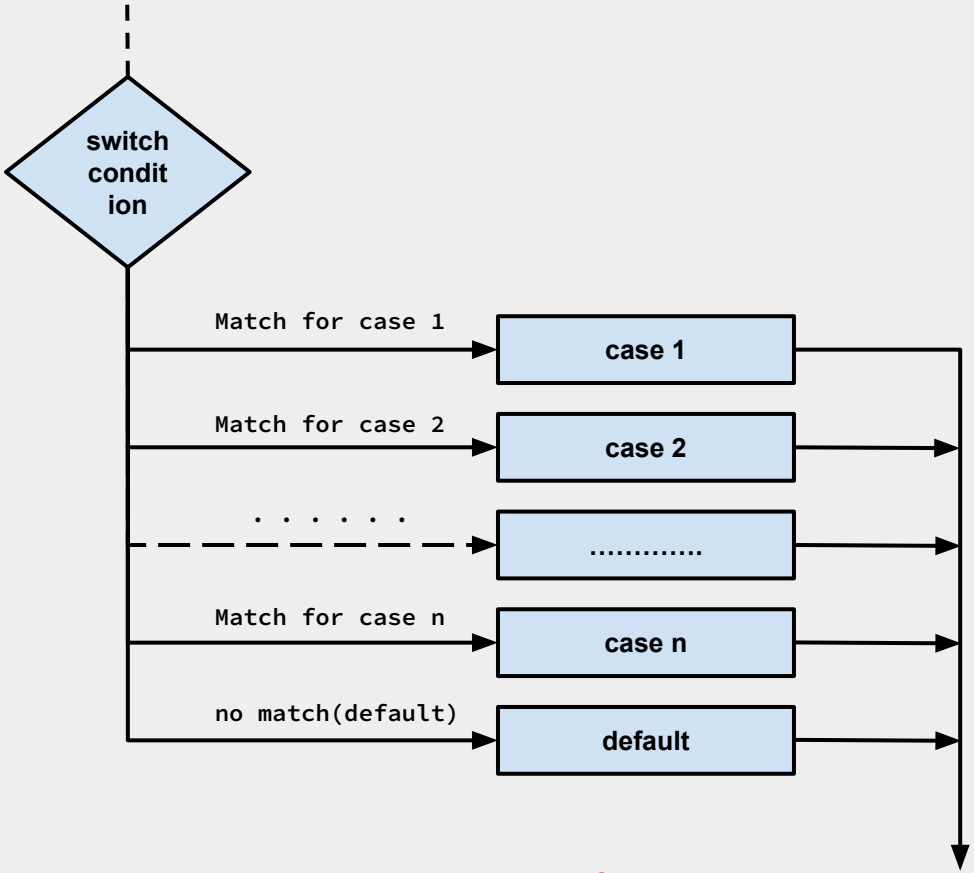
if-else


condition
true
false
Else statement
conditional code
Continue flow

nested if


condition
false
true
Statement 1
condition
false
true
Statement 2
condition
false
true
Statement 3
condition
true
false
Statement 4
Statement 5

switch case

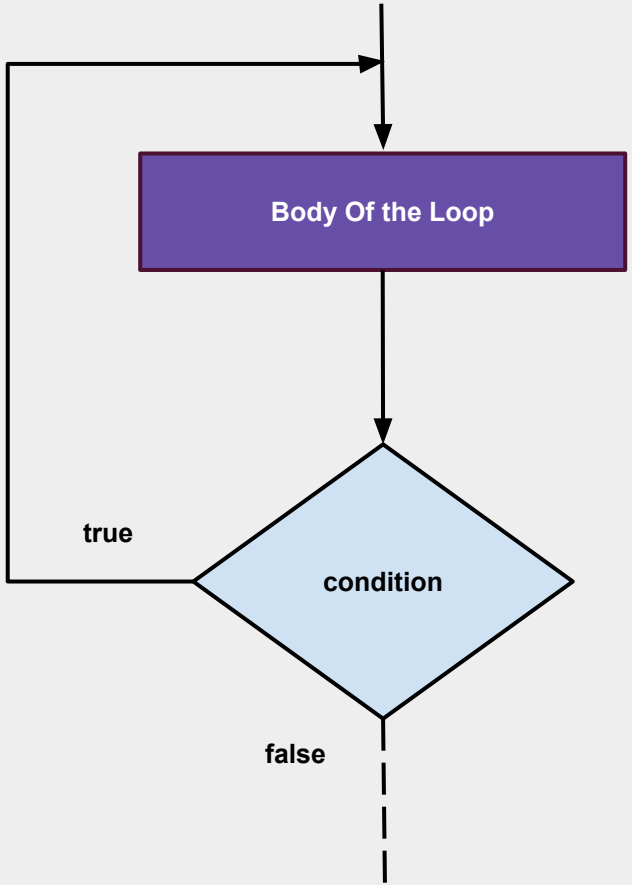


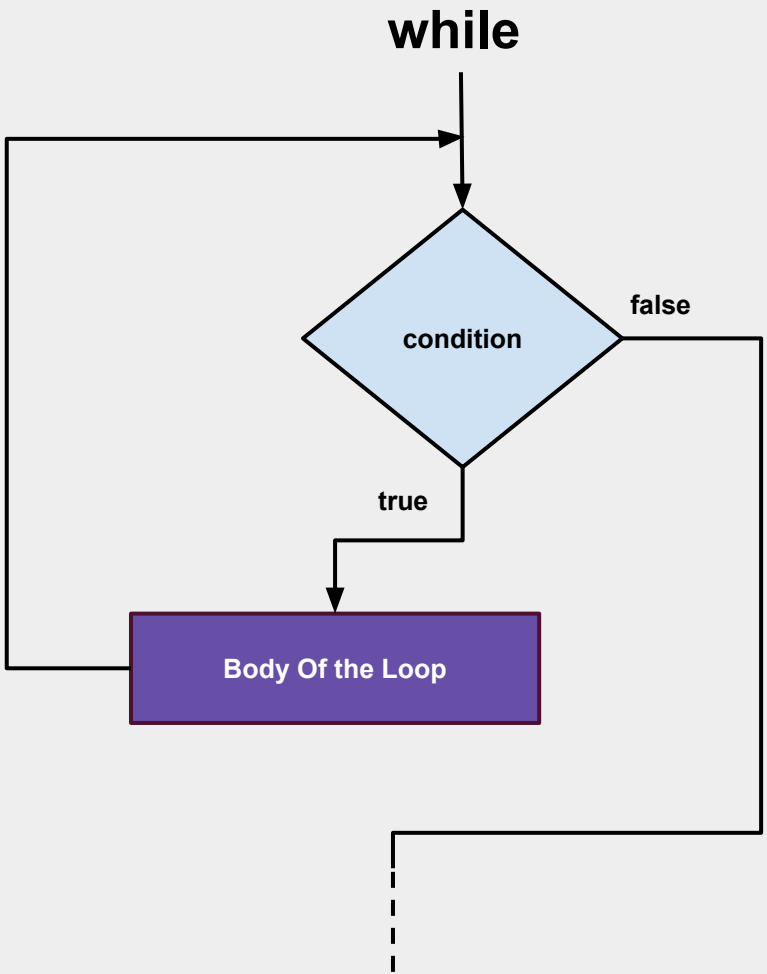
Note: Value can not be long, float, double, boolean.

Switch case and if-else difference

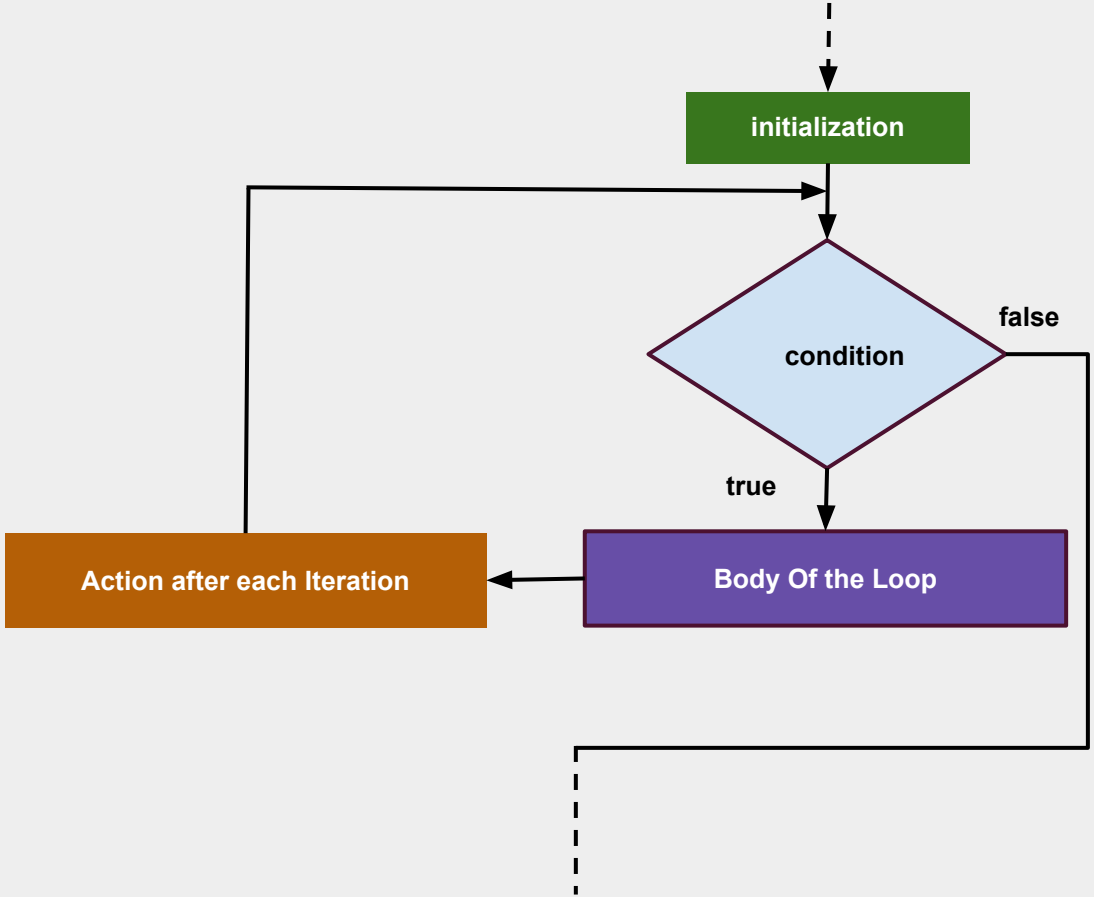
If-else statement	switch case statement
Here we need to use multiple statements for numerous decisions.	Here, we need to use a single statement for numerous decisions.
The if-else statement estimates integers, characters, floating points, and boolean types.	The switch statement estimates integers and character expressions.
In if-else, the values are based on conditions.	In the switch case, the values are based on user preference.
It is tough to edit if-else statements.	It is easy to modify the switch case.

do-while




while
condition
false
true
Body Of the Loop

for


initialization
condition
false
true
Body Of the Loop
Action after each Iteration

Infinite loops

- Infinite loops with while()
- Infinite loops with do-while()
- Infinite loops with for()

Unconditional statements

The **break** statement in Java terminates the loop immediately, and the control of the program moves to the next statement following the loop.

The **continue** keyword is used to end the current iteration in a for loop (or a while loop), and continues to the next iteration;

In Java programming, the **return** statement is used for returning a value when the execution of the block is completed. The return statement inside a loop will cause the loop to break and further statements will be ignored by the compiler.

label

- labelled loop
- labelled break
- labelled continue

Guess Game

Interview Questions

Stay hungry. Stay foolish