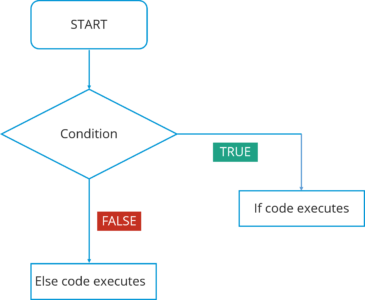


Now these statements can be further classified into the following:

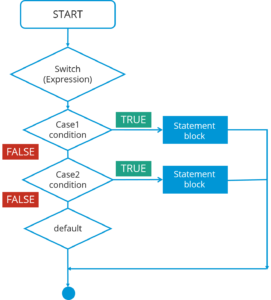
* If-else Statements
* Switch Statements

Refer to the following flowchart to get a better understanding of if-else statements:



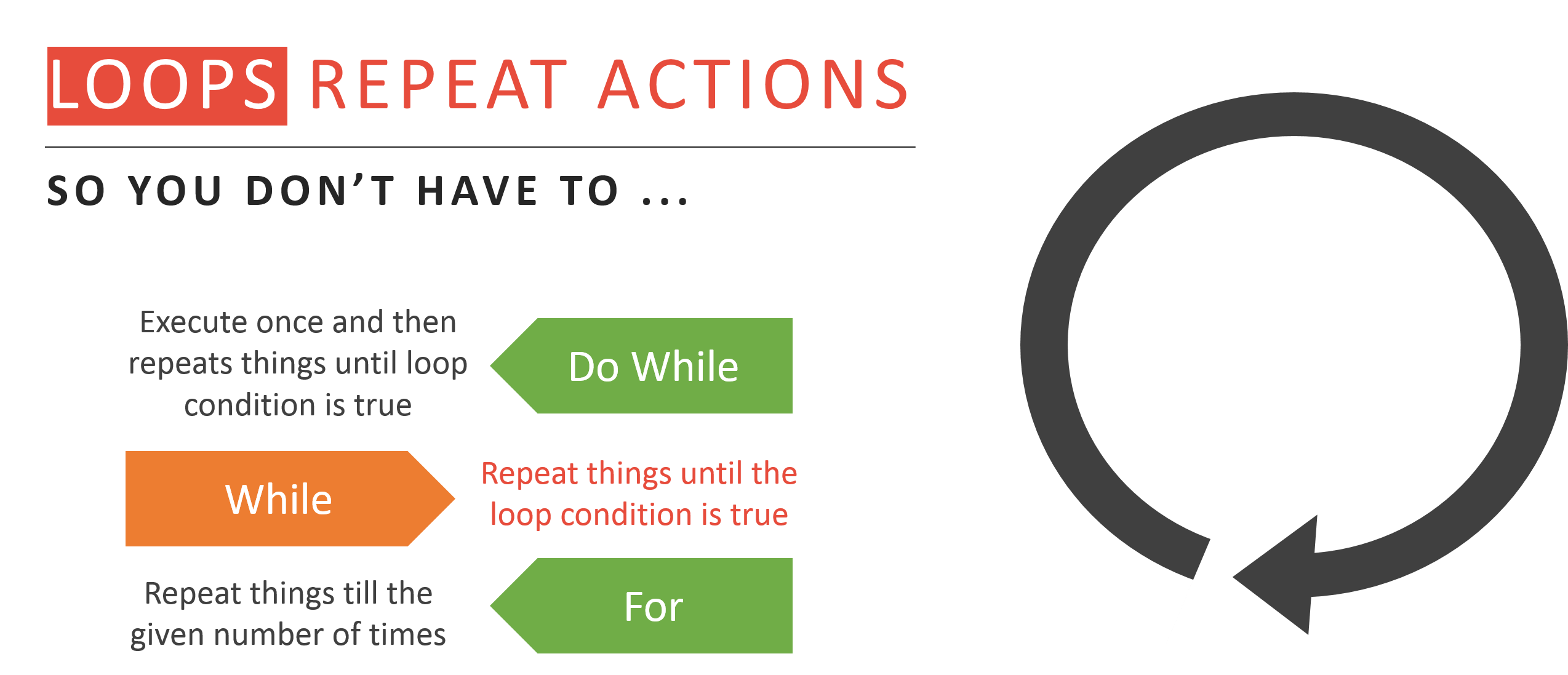
Moving on, we have **Switch case** statement. The switch statement defines multiple paths for execution of a set of statements. It is a better alternative than using a large set of if-else statements as it is a multi-way branch statement.

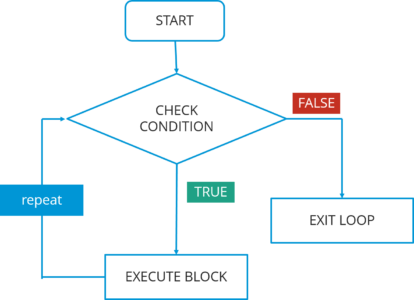
Refer to the following flowchart to get a better understanding of switch statements:

In this Switch case flowchart, the code will respond in the following steps:

1. First of all it will enter the switch case which has an expression.
2. Next it will go to Case 1 condition, checks the value passed to the condition. If it is true, Statement block will execute. After that, it will break from that switch case.
3. In case it is false, then it will switch to the next case. If Case 2 condition is true, it will execute the statement and break from that case, else it will again jump to the next case.
4. Now let’s say you have not specified any case or there is some wrong input from the user, then it will go to the default case where it will print your default statement.

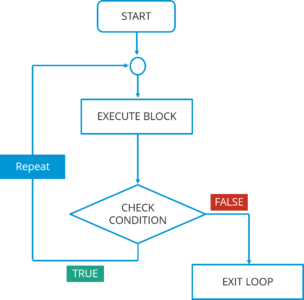
**Iteration Statements:**In Java, these statements are commonly called as loops, as they are used to iterate through small pieces of code. Iteration statements provide the following types of loop to handle looping requirements.



* **While statement: Repeat a group of statements while a given condition is true. It tests the condition before executing the loop body. Let’s understand this better with a flow chart:**  
  In this flowchart, the code will respond in the following steps:

1. First of all, it will enter the loop where it checks the condition.
2. If it’s true, it will execute the set of code and repeat the process.
3. If it’s False, it will directly exit the loop.

* **Do-while statement:** It is like a while statement, but it tests the condition at the end of the loop body. Also, it will executes the program at least once. Let’s understand this better with a flow chart:

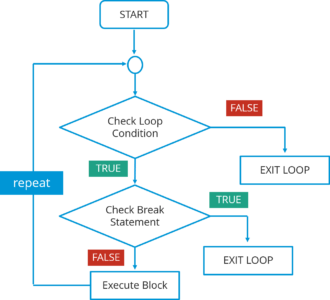


In this do-while flowchart, the code will respond in the following steps:

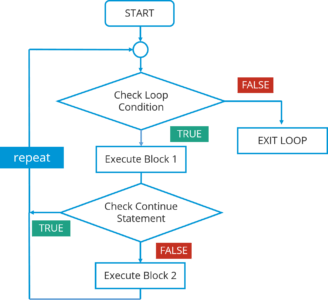
1. First of all, it will execute a set of statements that is mentioned in your ‘do’ block.
2. After that, it will come to ‘while’ part where it checks the condition.
3. If the condition is true, it will go back and execute the statements.
4. If the condition is false, it will directly exit the loop.

**Jump statement:** Jump statement are used to transfer the control to another part of your program. These are further classified into – *break* and *continue*.

Let’s learn about them in detail:

1. **Break statement:** Whenever a break statement is used, the loop is terminated and the program control is resumed to the next statement following the loop. Let’s understand this better with a flow chart:  
     
   In this flowchart, the code will respond in the following steps:  
   1. First of all, it will enter the loop where it checks the condition.  
   2. If the loop condition is false, it directly exits the loop.  
   3. If the condition is true, it will then check the break condition.  
   4. If break condition is true, it exists from the loop.  
   5. If the break condition is false, then it will execute the statements that are remaining in the loop and then repeat the same steps.

The syntax for this statement is just the ‘break’ keyword followed by a semicolon.

**2. Continue statement:**Continue statement is another type of control statements. The continue keyword causes the loop to immediately jump to the next iteration of the loop. Let’s understand this better with a flow chart:  


In this flowchart, the code will respond in the following steps:

1. First of all, it will enter the loop where it checks the condition.
2. If the loop condition is false, it directly exits the loop.
3. If the loop condition is true, it will execute block 1 statements.
4. After that it will check for ‘continue’ statement. If it is present, then the statements after that will not be executed in the same iteration of the loop.
5. If  ‘continue’ statement is not present, then all the statements after that will be executed.