**Q.3.Looping Concept,Break,Continue and return**

There may be a situation when you need to execute a block of code several number of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.

A **loop** statement allows us to execute a statement or group of statements multiple times.

**While loop:**

A **while** loop statement in Java programming language repeatedly executes a target statement as long as a given condition is true.

## Syntax

The syntax of a while loop is −

while(Boolean\_expression) {

// Statements

}

Here, **statement(s)** may be a single statement or a block of statements. The **condition** may be any expression, and true is any non zero value.

When executing, if the *boolean\_expression* result is true, then the actions inside the loop will be executed. This will continue as long as the expression result is true.

When the condition becomes false, program control passes to the line immediately following the loop.

**For loop:**

A **for** loop is a repetition control structure that allows you to efficiently write a loop that needs to be executed a specific number of times.

A **for** loop is useful when you know how many times a task is to be repeated.

## Syntax

The syntax of a for loop is −

For(initialization;Boolean\_expression;update)

{

//statement

}

Here is the flow of control in a **for** loop −

The **initialization** step is executed first, and only once. This step allows you to declare and initialize any loop control variables and this step ends with a semi colon (;).

Next, the **Boolean expression** is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop will not be executed and control jumps to the next statement past the for loop.

After the **body** of the for loop gets executed, the control jumps back up to the update statement. This statement allows you to update any loop control variables. This statement can be left blank with a semicolon at the end.

The Boolean expression is now evaluated again. If it is true, the loop executes and the process repeats (body of loop, then update step, then Boolean expression). After the Boolean expression is false, the for loop terminates.

**Do…while:**

A **do...while** loop is similar to a while loop, except that a do...while loop is guaranteed to execute at least one time.

## Syntax

Following is the syntax of a do...while loop −

do

{

//Statement

}while(Boolean\_expression);

Notice that the Boolean expression appears at the end of the loop, so the statements in the loop execute once before the Boolean is tested.

If the Boolean expression is true, the control jumps back up to do statement, and the statements in the loop execute again. This process repeats until the Boolean expression is false.

**Break:**

The **break** statement in Java programming language has the following two usages −

When the **break** statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.

It can be used to terminate a case in the **switch** statement (covered in the next chapter).

## Syntax

The syntax of a break is a single statement inside any loop −

Break;

**Continue:**

The **continue** keyword can be used in any of the loop control structures. It causes the loop to immediately jump to the next iteration of the loop.

In a for loop, the continue keyword causes control to immediately jump to the update statement.

In a while loop or do/while loop, control immediately jumps to the Boolean expression.

## Syntax

The syntax of a continue is a single statement inside any loop −

Continue;

**Return:**

 The **return**statement is used to explicitly return from a method. That is, it causes program control to transfer back to the caller of the method. As such, it is categorized as a jump statement.

A method can return a value or reference type or does not return a value. If a method does not return a value, the method must be declared void and it doesn’t need to contain a return statement.

If a method declare to return a value, then it must use the return statement within the body of method. The data type of the return value must match the method’s declared return type.