3. Control Flow Statements.

1) If-Else Statements

Answer :- If- else together represents the set of Conditional statements in Java that are executed according to the condition which is true.

If the if Condition is True than the IF block will execute otherwise else Condition will execute.

Syntax:-

If(Condition)

{

    //Block of Code;

}

Else

{

   //Block of Code;

}

Example:-

If(n % 2 ==0)

{

   System.out.println("Number is Even");

}

Else

{

   System.out.println("Number is Odd");

}

TOPS Technology

2) Switch Case Statements

Answer:- The **switch statement in Java** is a multi-way branch statement. In simple words, the Java switch statement executes one statement from multiple conditions.

Syntax:-

Switch(variable)

{

Case 1: //Statement1;   break;

Case 2: //Statement2;   break;

Case 3: //Statement3;   break;

Case 4: //Statement4;   break;

Default : //Invalid Statement; break;

}

3) Loops (For, While, Do-While).

Answer :- In Java, loops are used to execute a block of code multiple times until a condition is met.

Two Types of Loops are there :-

Entry Control Loop :- For loop and While Loop.

Exit Control Loop :- Do _ while Loop.

For Loop :-

The for loop is used when the number of iterations is known. It has three parts:

Initialization , Condition, Increment / decrement .

Syntax :-

For(Initialization ; Condition ; Increment / Decrement)

{

```
    // Block of Code;

}
```

While Loop :-

The while loop repeats a block of code as long as a specified condition is true. It's used when the number of iterations isn't known in advance.

Syntax :-

While(Condition)

```
{

    // Block of Code;

    Increment / decrement;

}
```

4) Break and Continue Keywords

Answer :- In Java, the break and continue keywords are used to control the flow of loops by altering how and when they proceed.

Break Example :-

The break keyword is used to immediately terminate a loop (or switch statement).

```
for (int i = 0; i < 10; i++) {

  if (i == 5) {

    break;  // Exit the loop when i equals 5

  }

  System.out.println("Iteration " + i);

}
```

Continue Example:-

The continue keyword skips the current iteration and moves to the next iteration in the loop.

```
for (int i = 0; i < 10; i++) {

    if (i % 2 == 0) {

        continue;  // Skip even numbers

    }

    System.out.println("Iteration " + i);

}
```