Lecture 15 & 16

Control-flow Statements

- Control flow statements are used to control / manage the flow of execution of a program based on certain conditions.
- By default control structure or flow of program is sequential i.e. ~~compiler exe~~ JVM executes the code from top to bottom line-by-line.

  e.g:- Sum of 2 numbers program.

  entry point is main & then statements would be executed sequentially.

- Control flow statements gives us the ability to manage the program's flow, without these we would have no control over the sequence or flow.

  Java provides 3 types of control flow statements:-

  ① Decision-making (if-else, switch)
  ② Looping (for, while, do-while)
  ③ Jump (break, continue, return)

① **Decision-making:-** These statements let the program decide which path to take on conditions.

→ Decision-making statements evaluates the given condition / boolean expression and control the program flow depending upon the result of the condition provided.

→ Java provides 2 types of decision-making statements:-

  ① → if
  ② → Switch

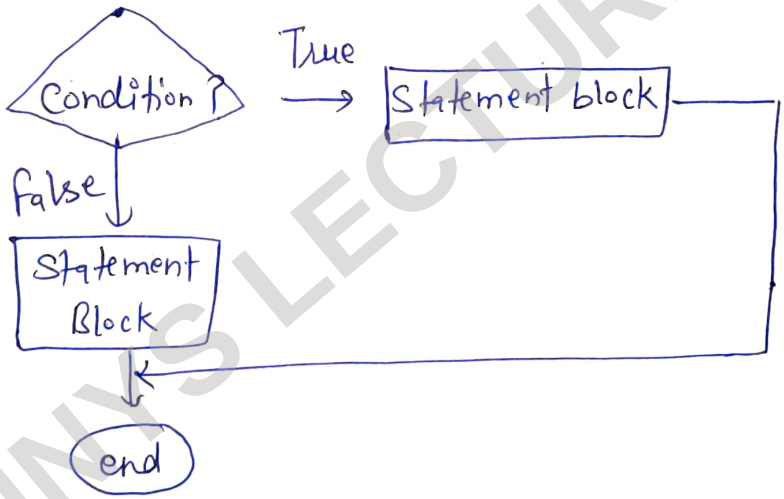
(1) if Statements :-

↳ it is the simplest decision-making statement

Syntax :-

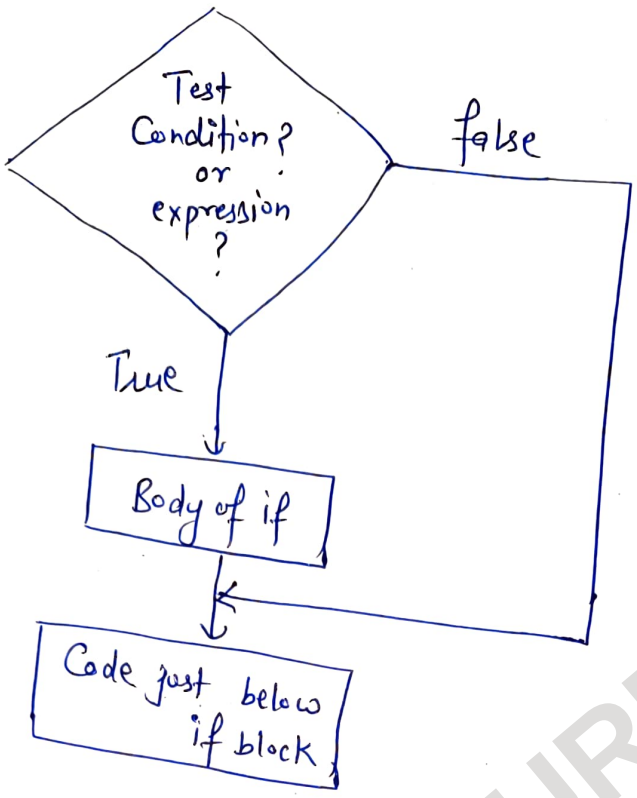
```
if (condition) {
    // Code in this block only executes if the
    // condition is true
}
```

How to select or decide based on some condition. General Syntax | flow chart is :-



e.g :- if green light then go
if red light then stop

Working of if :- The given condition will be evaluated either true or false. If true then a block of code would be executed otherwise it will ignore it & skip to the the statement just below the if block.



e.g:-

```
int temperature = 42;
if (temperature >= 40) {
    System.out.print("Heatwave Alert! Stay Indoors.");
}
System.out.print("\n Outside If Block!");
```

NOTE:- The condition in if must be a boolean expression or must evaluate to true or false (boolean values) otherwise it will give compilation error.

e.g:- in above example if we write:-

```
if (temperature) {          → will give error.
    System.out.print("Heatwave Alert!");
}
```

Here, `int temperature = 42;` → this complete is a statement

& `temperature = 42` → is an expression

If we write `int temperature = 42;` [whitespaces are there]

→ this will work fine. whitespaces are ignored by Java

Practice Time:-

① `if temperature >=40 {` → error, brackets are required.

```
    - - -
}
```

②
```
int a, b;
a = 5; b = 10;
if ( a+b > 12)
    System.out.println ("Inside If");
System.out.println ("Outside if block");
```

IMP Point

No need of parenthesis {} if we have only one statement within if block.

③
```
if (a+b >12); → no error.
{
    // Code block
}
```

④
```
if ( a+b >18)
{   System.out.println ("Hi");
    System.out.println ("Hello");
System.out.println ("Outside If");
```

⇒ Output:-
Hello
Outside If

Coding Exercise:-

① WAP to check if number is positive, negative or zero.

② WAP to check that a given number is even, positive or zero.

③ WAP to check if a number is positive.

④ WAP to check if the temperature (in celsius) is above freezing point (0°C).

⑤ WAP to check if you can successfully withdraw the money / if you have sufficient balance to withdraw money in your bank account.

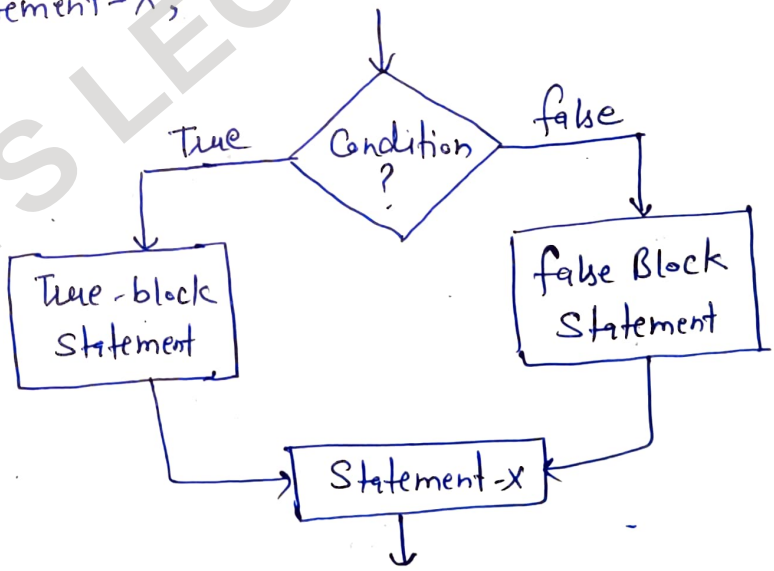
if-else Statement :- It is extension of simple if.

if-statement only tells what to do when the condition evaluates to true, it doesn't tell anything when condition is false.

- if-else statement provides an else block. The else block is executed when the condition of if-block is evaluated as false.

General form :-

```
if (Condition)
{
    // True-block statement(s)
}
else {
    // False-block statement(s)
}
Statement-X;
```



→ It is double-selection statement because it selects between two different actions.

→ At one time either true-block or false-block will be executed, not both. and in both the cases control will be transferred subsequently to statement-X.

NOTE:- else without if block is not possible.

→ In else-block if we have only one statement to execute then no need to put parantheis ({ }).

Coding Exercise :-

1. WAP to check if number is positive or negative.
2. WAP to check/find maximum of 2 numbers.
3. WAP to check number is even or odd.

if with an else if / ~~Else~~ else-if ladder / if-else-if ladder :→

→ If you want to test multiple conditions then we use else-if ladder.

→ It is known as multiple selection statement.

Syntax:-

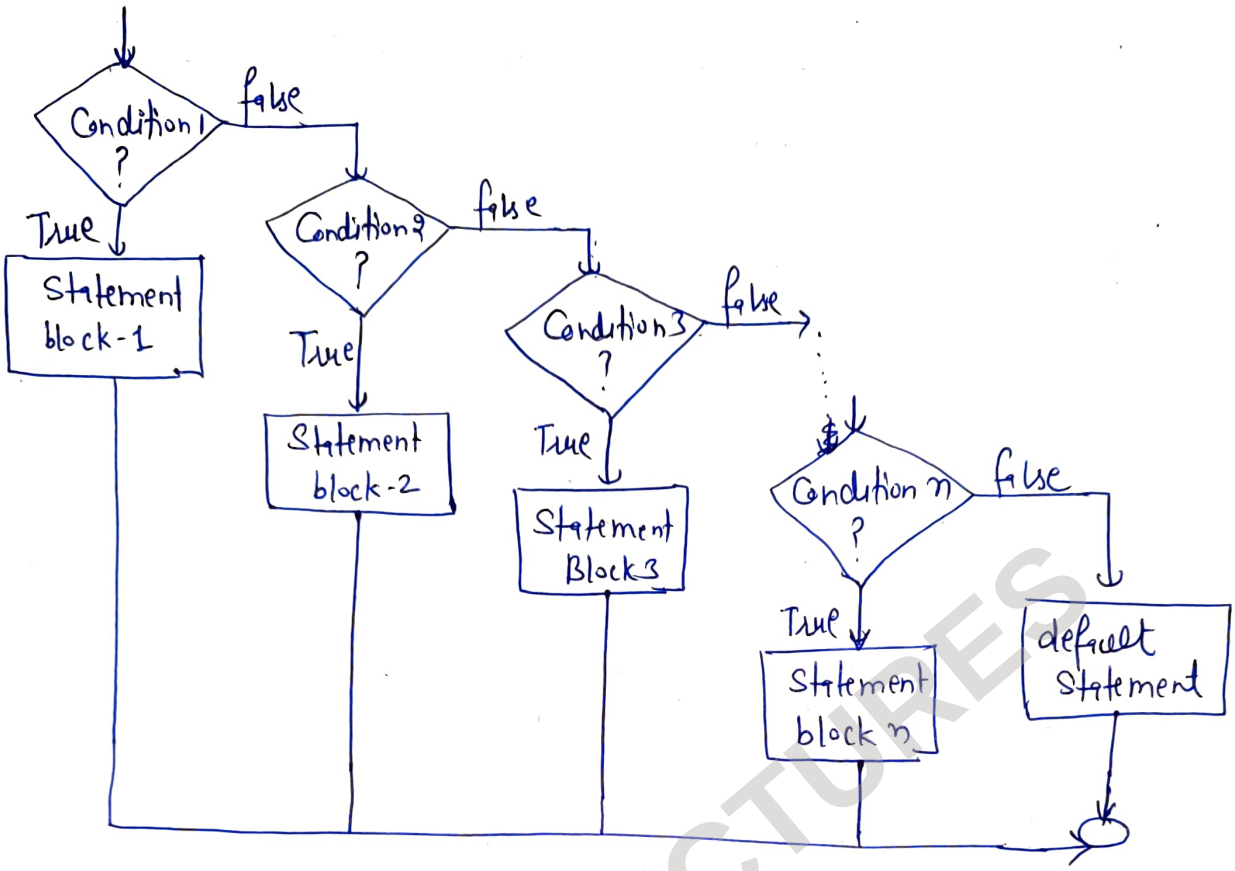
```
if (Condition1) {
    // Code in block will execute only if first condition
    // is true
} else if (Condition 2) {
    // Code in this block will execute only if Condition 2
        is true.
}

else {
    // Code in this block will execute only if all the
    // above conditions are false.
}
```

NOTE:- → This else block must be last and it is optional.

There can be as many else-if block as you want but else block would be the last. After else you can not write any else-if block.

- The conditions are evaluated from top to bottom. As soon as a true condition is found the statement associated with it is executed, skipping the rest of the else-if ladder.



If all the conditions are evaluated to false, then final else block containing default statement will be executed.

Coding Exercise:-

① WAP to check if number is +ve, -ve or zero.

② WAP that asks the user to input a score (0 to 100) and evaluates the grade based on the following conditions:-

90-100 :- Grade A
80-89 :- Grade B
70-79 :- Grade C
60-69 :- Grade D
Below 60 :- Fail

If score is out of range (less than 0 or more than 100) then display a message:- Invalid score. Please enter a score between 0 and 100

Nested-if Statement :- if inside another if statement.

- Java allows a if or if-else statement inside another if or else-if or else statement.

Syntax :-

```
if ( Condition 1 ) {

    if ( Condition 2 ) {
       // executes if both Condition 1 and Condition 2
       //          are true
     } else {
      // executes if condition 1 is true but condition 2 is
       //  false
    }
} else {

    //executes if condition 1 is false

}
```

→ Here if the condition1 evaluates to false then else block will be executed & skips the nested if-else block.

e.g:-

```
int age = 23, weight = 47;
if (age >= 18) {
   System.out.println(" You can vote ! ");
   if ( weight >= 45) {
      System.out.println (" You are fit to donate Blood ! ");
    } else {
      System.out.println (" You can not donate blood ! ");
    }
} else {
   System.out.println (" You are not eligible to cast vote! ");
}
```

Coding Exercise on Nested if

① WAP to check whether a student is eligible for a scholarship based on three criteria:

1. Must have atleast 85% marks
2. Must have participated in at least one event.
3. Family income should be less than or equal to 8,00,000 per year.

[if marks are atleast 85% then only check for event participation & family income]

Coding Exercise on elif / else-if ladder :-

① WAP for BMI Calculator with Interpretations

Underweight = < 18.5
Normal weight = 18.5 - 24.9
Overweight = 25 - 29.9
Obesity = BMI of 30 or greater.

$$BMI = \frac{weight\ (kg)}{height\ (meter)^2}$$

⇒ Print the appropriate message according to the BMI value.

e.g. if bmi is under 18.5 (not including) print "You are Underweight"