



GALGOTIAS
UNIVERSITY

**School of Computing
Science and Engineering**

Program: BCA - IOP

Course Code: BCAS3031

Course Name: PL/SQL & Cursors and
Triggers

Dr. T. Poongodi
Associate Professor

Decision Making in PL/SQL (if-then , if-then-else, Nested if-then, if-then-elsif-then-else)

Decision-making statements in programming languages decide the **direction of flow of program execution**. Decision-making statements available in pl/SQL are:

- if then statement
- if then else statements
- nested if-then statements
- if-then-elsif-then-else ladder

if then statement

if then statement is the most simple decision-making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statement is executed otherwise not.

Syntax:

```
if condition then  
-- do something  
end if;
```

Here, condition after evaluation will be either true or false. if statement accepts boolean values – if the value is true then it will execute the block of statements below it otherwise not. if and endif consider as a block here.

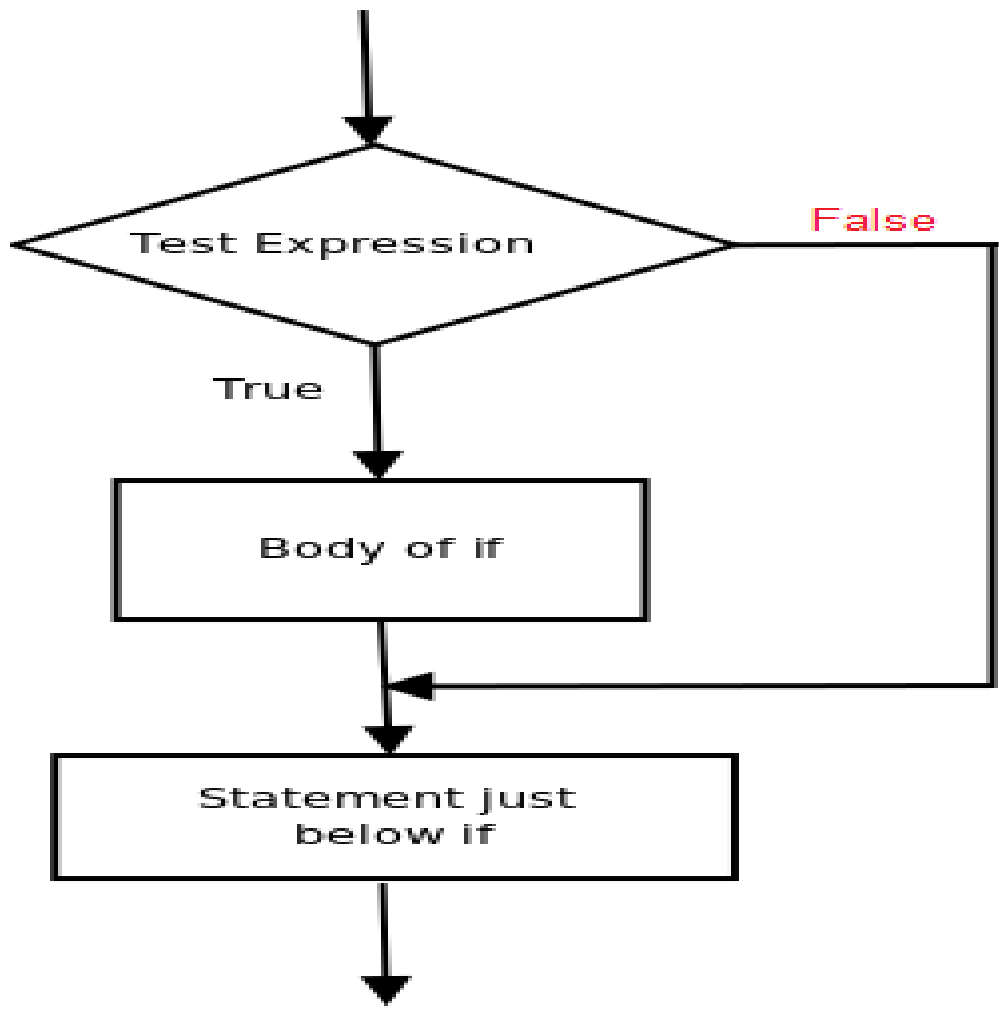
```
declare
-- declare the values here

begin

if condition then
dbms_output.put_line('output');

end if;

dbms_output.put_line('output2');
end;
```



```
-- pl/sql program to illustrate If statement
declare
num1 number:= 10;
num2 number:= 20;

begin

if num1 > num2 then
dbms_output.put_line('num1 small');
end if;

dbms_output.put_line('I am Not in if');

end;
```

As the condition present in the if statement is false.
So, the block below the if statement is not executed.
Output:

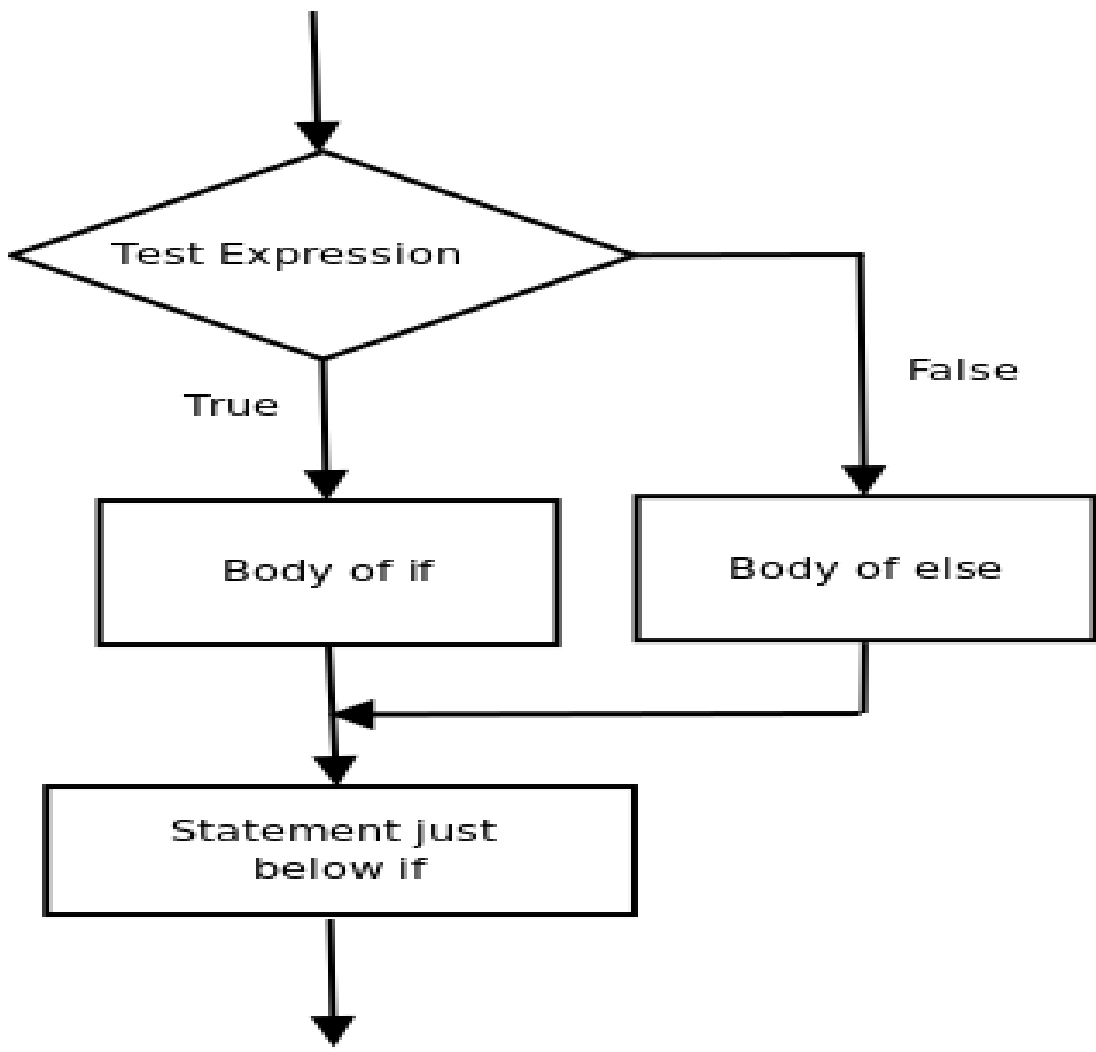
```
I am Not in if
```


if – then- else:

The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won't.

Use the else statement with if statement to execute a block of code when the condition is false.

```
if (condition) then
-- Executes this block if
-- condition is true
else
-- Executes this block if
-- condition is false
```



```
-- pl/sql program to illustrate If else statement
declare
num1 number:= 10;
num2 number:= 20;

begin

if num1 < num2 then
dbms_output.put_line('i am in if block');

ELSE
dbms_output.put_line('i am in else Block');
end if;

dbms_output.put_line('i am not in if or else Block');
end;
```

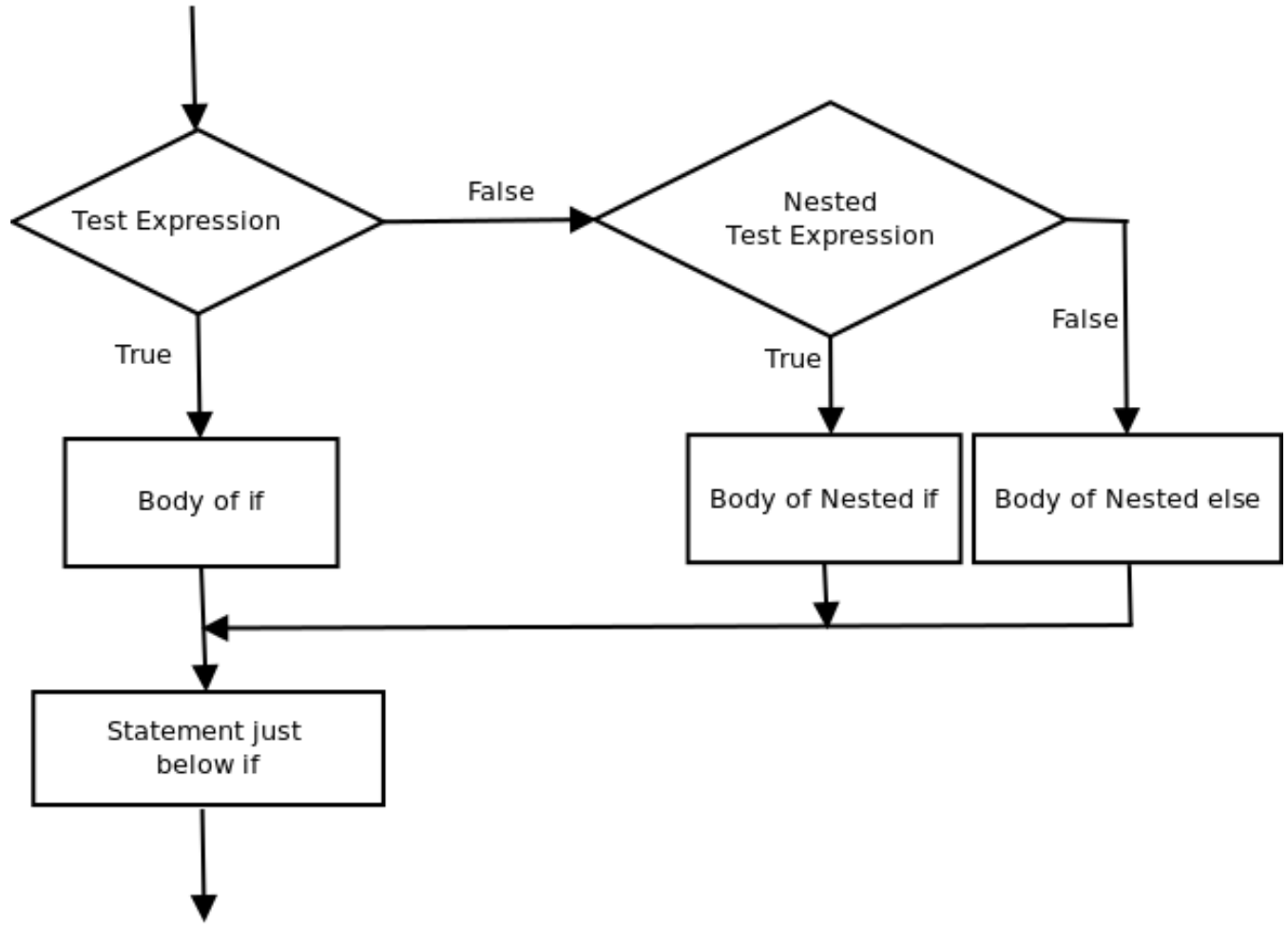
The block of code following the else statement is executed as the condition present in the if statement is false after calling the statement which is not in block(without spaces).

```
i'm in if Block  
i'm not in if and not in else Block
```

nested-if-then:

A nested if-then is an if statement that is the target of another if statement. Nested if-then statements mean an if statement inside another if statement.

```
if (condition1) then
    -- Executes when condition1 is true
    if (condition2) then
        -- Executes when condition2 is true
    end if;
end if;
```




```
-- pl/sql program to illustrate nested If statement
declare
num1 number:= 10;
num2 number:= 20;
num3 number:= 20;

begin
if num1 < num2 then
dbms_output.put_line('num1 small num2');

    if num1 < num3 then
dbms_output.put_line('num1 small num3 also');
end if;

end if;

dbms_output.put_line('after end if');
end;
```

```
num1 small num2  
num1 small num3 also  
after end if
```

if-then-elsif-then-else ladder

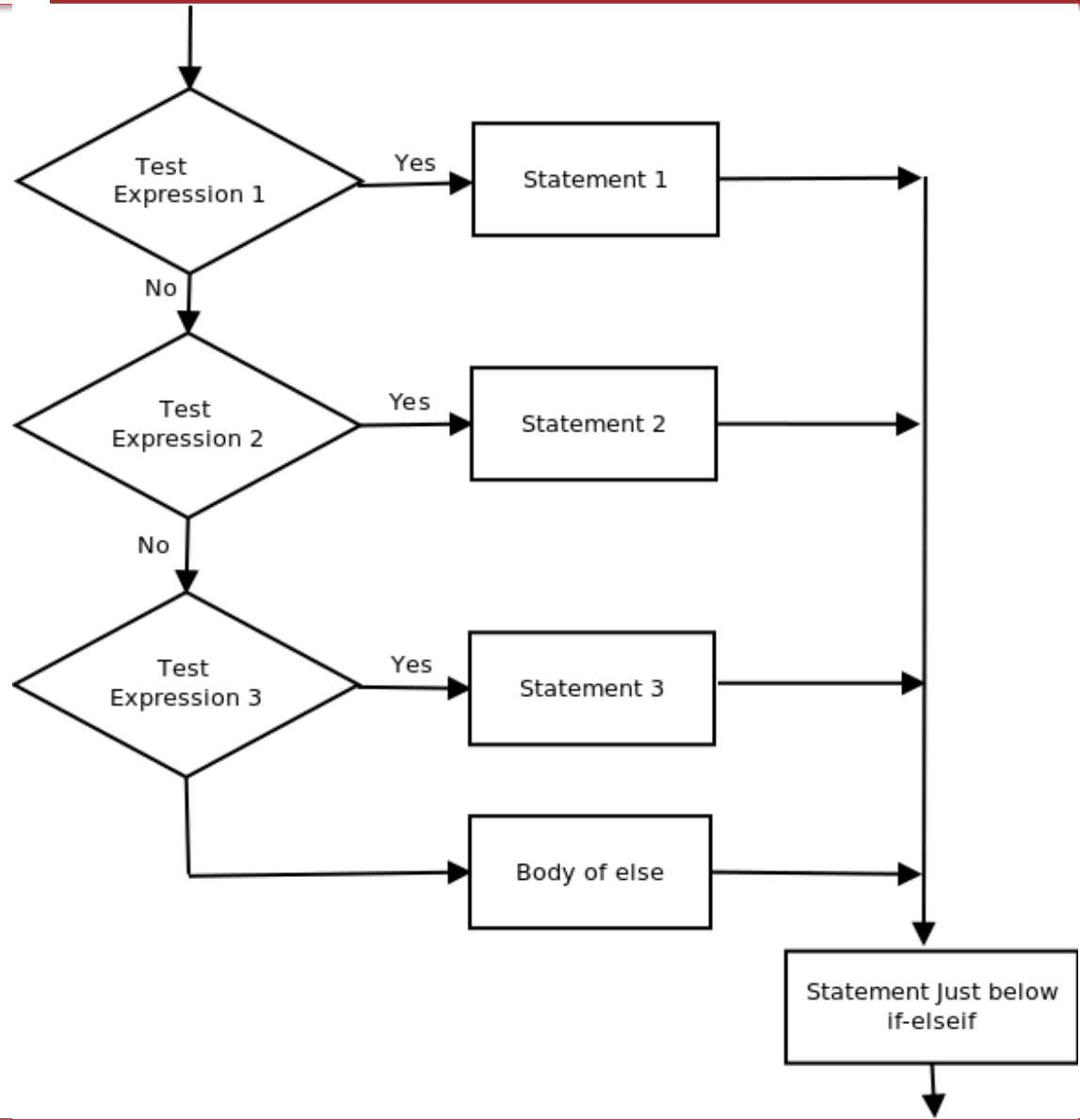
Here, a user can decide among multiple options.

The if then statements are executed from the top down.

As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the ladder is bypassed.

If none of the conditions is true, then the final else statement will be executed.

```
if (condition) then
    --statement
elsif (condition) then
    --statement . .
else
    --statement
endif
```





```
-- pl/sql program to illustrate if-then-elif-then-else ladder
declare
num1 number:= 10;
num2 number:= 20;

begin

if num1 < num2 then
dbms_output.put_line('num1 small');

ELSIF num1 = num2 then
dbms_output.put_line('both equal');

ELSE
dbms_output.put_line('num2 greater');
end if;

dbms_output.put_line('after end if');
end;
```

```
num1 small  
after end if
```

```
DECLARE
a NUMBER :=10;
BEGIN
dbms_output.put_line('Program started.' );
IF( a > 100 )
THEN
dbms_output.put_line('a is greater than 100');
END IF;
dbms_output.put_line('Program completed.');
```

```
END;
/
```


Program started.
Program completed.

```
DECLARE
a NUMBER:=11;
BEGIN
dbms_output.put_line ('Program started');
IF( mod(a,2)=0) THEN
dbms_output.put_line('a is even number' );
ELSE
dbms_output.put_line('a is odd number1);
END IF;
dbms_output.put_line ('Program completed.');
```

```
Program started.  
a is odd number  
Program completed.
```

Nested- If Statement: Greatest of three number

```
DECLARE
mark NUMBER :=55;
BEGIN
dbms_output.put_line('Program started.' );
IF( mark >= 70) THEN
dbms_output.put_line('Grade A');
ELSIF(mark >= 40 AND mark < 70)
THEN
dbms_output.put_line('Grade B');
ELSIF(mark >=35 AND mark < 40) THEN
dbms_output.put_line('Grade C');
END IF;
dbms_output.put_line('Program completed. ');
END;
/
```

```
/*Nested-if2 */
    dbms_output.put_line('Checking Nested-IF 2' );
    IF( b > c ) THEN
        dbms_output.put_line('B is greatest' );
    ELSE
        dbms_output.put_line('C is greatest' );
    END IF;
END IF;
dbms_output.put_line('Program completed.' );
END;
/
```



Thank You