DECISION / CONTROL STATEMENT:

- The decision statement helps the programmer to skip the block of instructions from the execution if the condition is not satisfied.
- **TYPES OF DECISION STATEMENTS:**
  - if statement
  - if-else statement
  - if-else if ladder
  - switch statement

Syntax to create if statement:

- if (condition)

```
{
    //statements
}
```

WORKFLOW:

- If the condition is satisfied then the instruction written inside the if block gets executed or normal flow of the execution continues(Instructions written inside the if block is skipped)

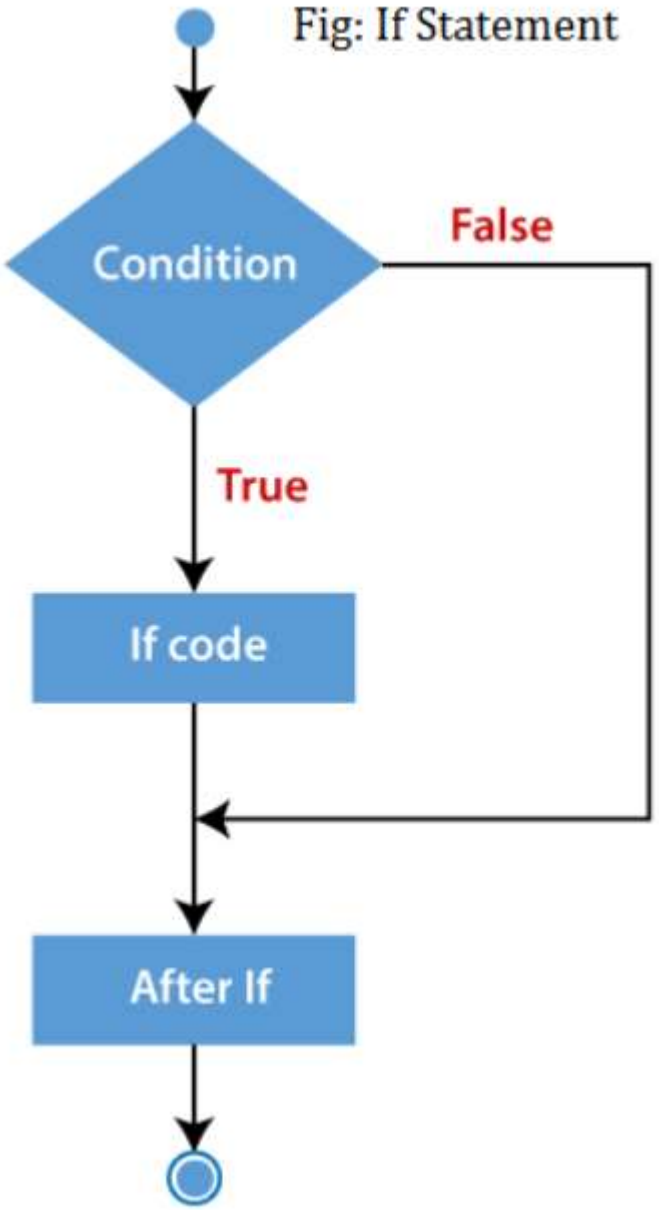


Fig: If Statement

Syntax to create if-else statement

- if (condition)
  {
  }

  else
  {
  }

Flowchart

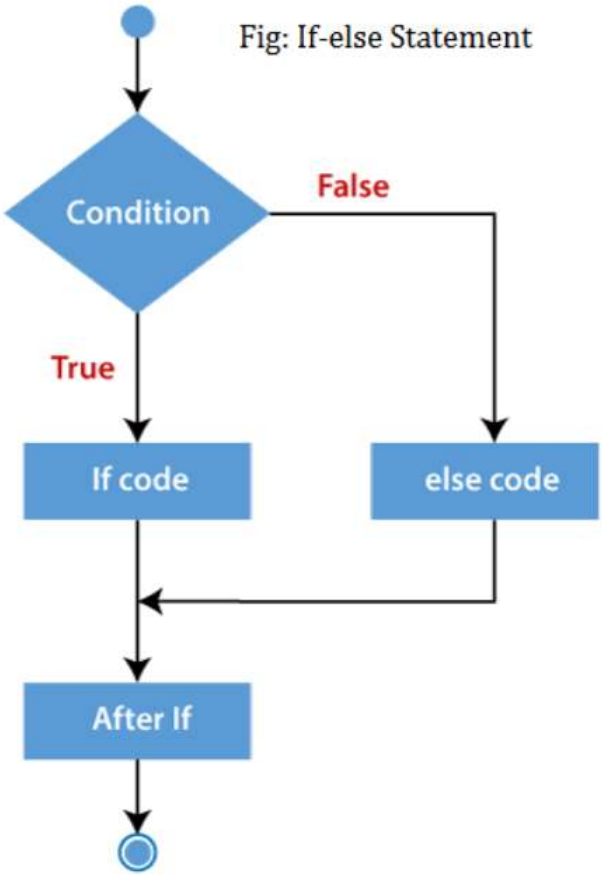


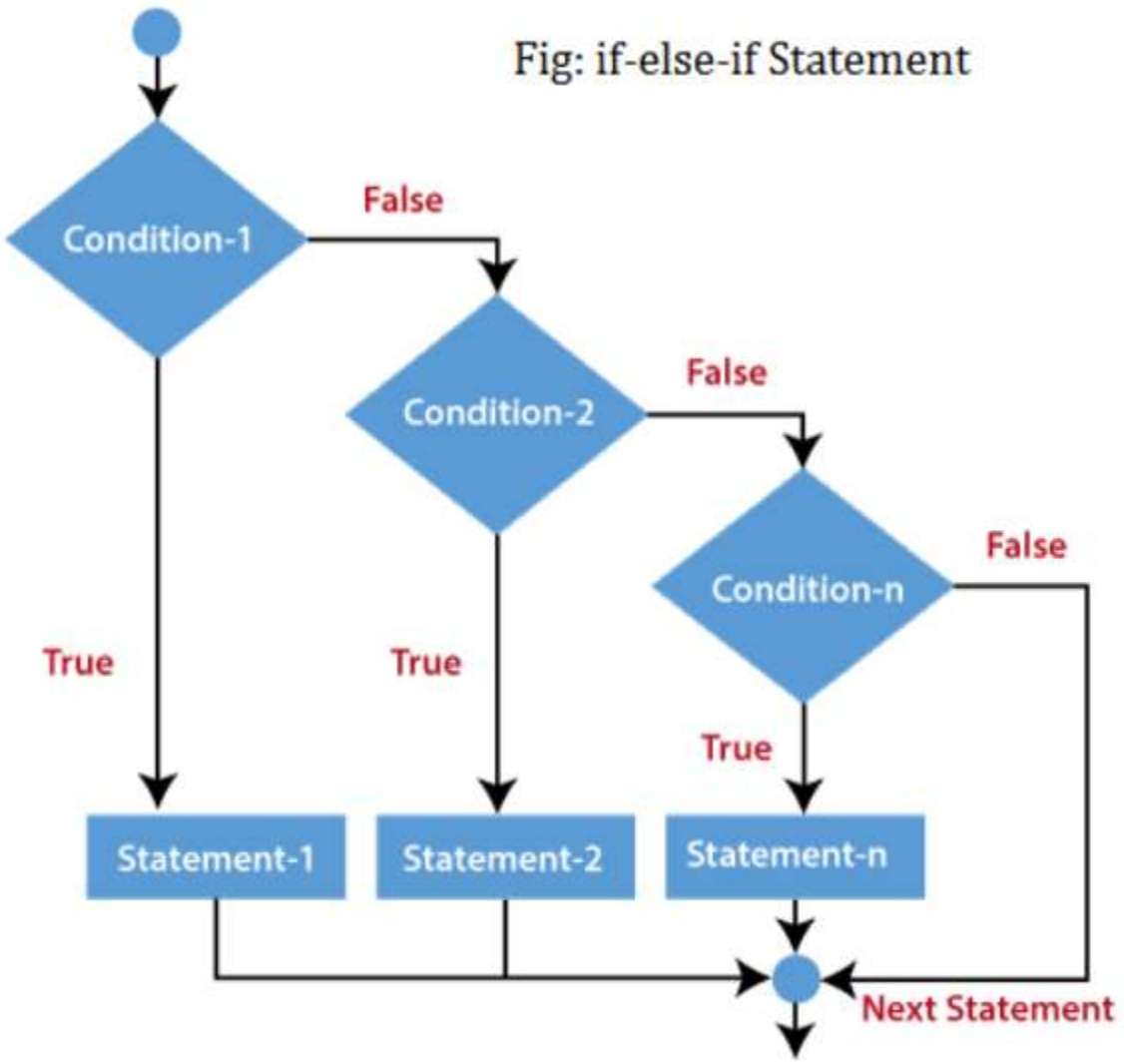
Fig: If-else Statement

WORKFLOW:

If the condition is satisfied then
the instruction written inside the
if block gets executed if not satisfied
else block gets executed (any one of the blocks will be skipped based on a condition.)

Syntax to create if-else if statement

```
if (condition)
{
}
else if (condition)
{
}
else if (condition)
{
}
else
{
}
```



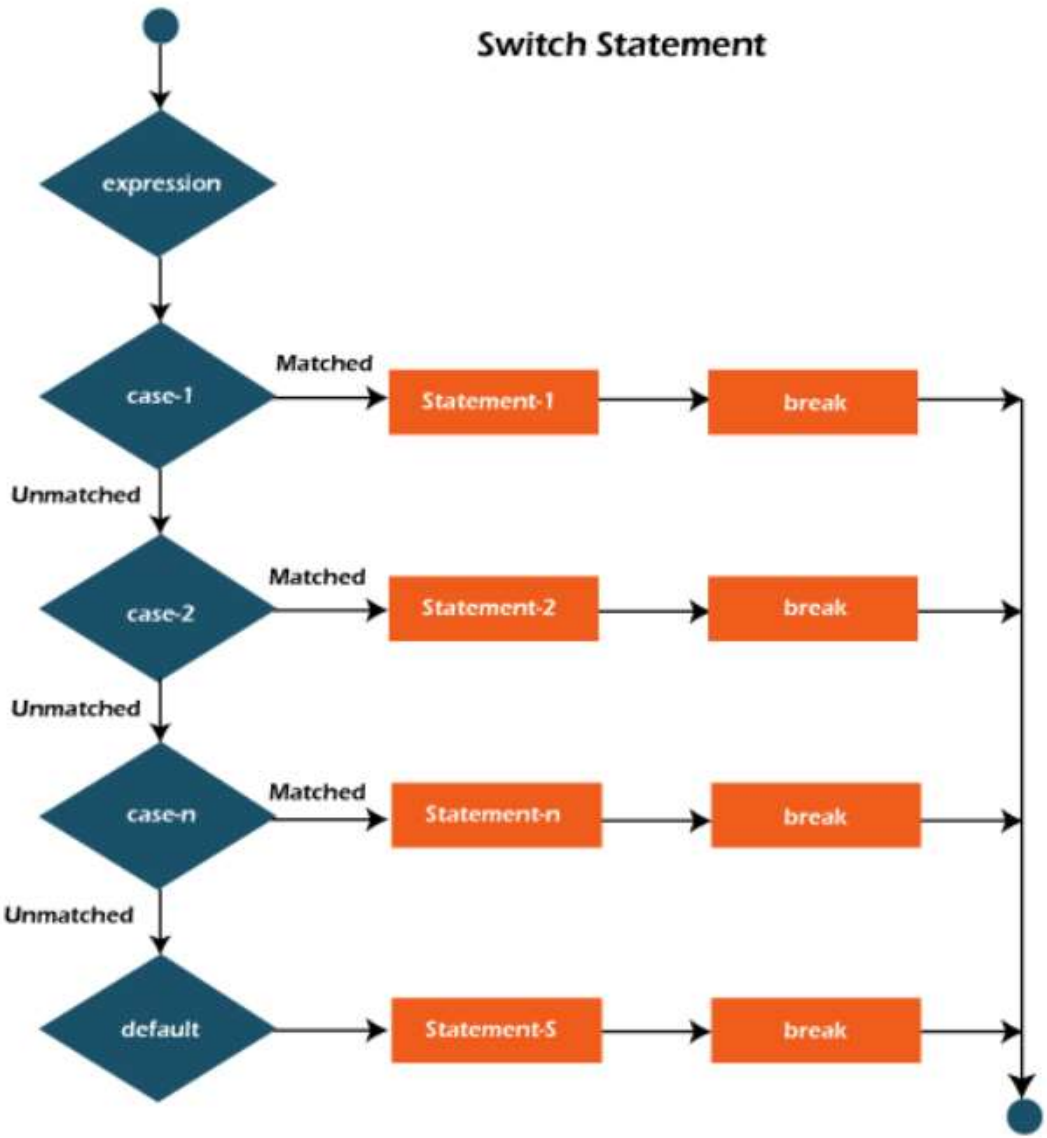
WORK FLOW of else if ladder:

If the condition is satisfied then the instruction written inside the if block gets executed if not satisfied, the condition is checked in the else if block from top to bottom order and if the condition is satisfied in any of the else if block then , only that else if block is gets executed if not satisfied else block gets executed remaining blocks are skipped

```
if (condition1)
{
//code to be executed if condition1 is true
}
else if (condition2)
{
//code to be executed if condition2 is true
}
 else if (condition3)
{
//code to be executed if condition3 is true
}
else
{
//code to be executed if all given conditions are false}
```

Syntax to create switch block

```
switch(value / variable / expression )
{
case value / expression :
{
   statement;
}
break ;
case value / expression :
{
   statement;
}
break ;
.
.
.
default :
{
   statement ;
}
break ;
}
```



- **WORKFLOW :**
  - The value / variable / expression passed in the switch gets compared with the value passed in the case from top to bottom order.
  - If any of a case is satisfied, the case block is executed and all the blocks present below get executed.
  - If no case is satisfied then the default block gets executed.
  - For a case we can use a break statement which is optional.
- **NOTE:**
  - **For a switch we can't pass long, float, double, boolean.**
  - **For a case we can't pass a variable.**
- **BREAK:**
  - break is a keyword, it is a control transfer statement.
  - break is used inside the switch and loop block
  - When the break statement is executed control is transferred outside the block.