

Learning Goals

[Click to Continue](#)



This session will help you understand,

- Basics of conditional and looping statements
- Conditional control structures
 - If
 - If-else
 - Switch



Statements and Blocks

Click to Continue



A *java statement* is a complete instruction terminated by a semi-colon.

Example:

```
String country="India";
```

A *block* is group of statements enclosed in curly brackets.

Example:

```
{  
    country="India";  
    temperature=23;  
}
```

Executed First

Executed Second

Each statement is execute one by one.



Control Statements

[Click to Continue](#)



The *control statements* allows programmers to control the flow of program's execution based on some conditions.



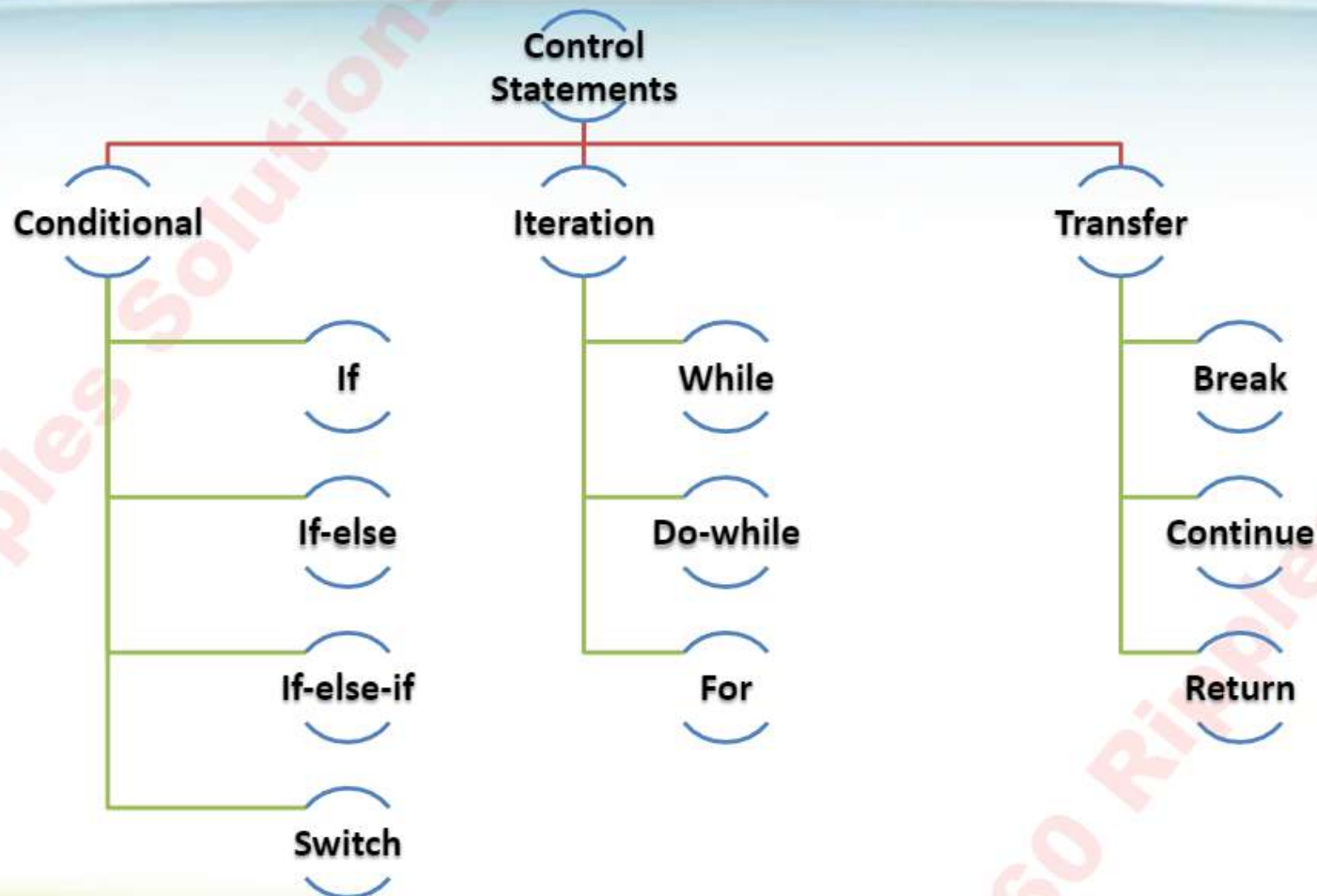
The control statements are classified as,

- *Repetitive execution – Executing a statement (or) block 'N' number of times*
- *Conditional execution– Execute statement or block based on some conditions.*



Control Statements Types

[Click to Continue](#)

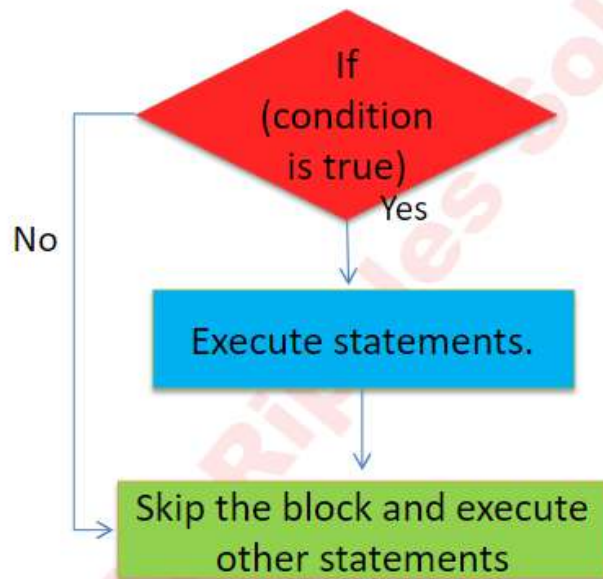


Simple If Statement

Click to Continue



The simple if statement allows the execution of a single statement or a block of statements based on a given condition.



Method I:

```
if(condition1 ) {  
    //statement body;  
}
```

Method II:

```
If(condition1 && condition2){  
    //statement body;  
}
```

← The condition can be a single relational expression.

← The condition can also be a combination of **more than one** relational expression separated using a logical operators



Illustration - If Statement

Click to Continue



Illustration I: Without curly braces

```
int x = 20;  
int y = 30;  
If(x>y)  
    System.out.println("Value of x is greater than y");  
    System.out.println("This is not part of IF condition");
```

Only this statement is executed based on the IF condition.

Illustration II: With curly braces

```
int studentId=900;  
int studentAge=18;  
int marks=80;  
If((studentAge<20) && (marks >70)){  
    System.out.println("Eligible for graduation");  
    System.out.println("Welcome");  
}
```

If both the condition **marks** and **age** both are met the print statements will be printed

Best Practise: It is a good programming practice to use the curly braces regardless of whether there is one or more statements.



If-else

Click to Continue



This is similar to a simple if statement with also an *else* block.

If-else will have two blocks of code, the if block and else block.

Syntax:

```
if( <condition1> ) {  
    statements  
}  
else{  
    statements  
}
```

If condition1 is **satisfied** the if block statements are executed

If condition1 is **not satisfied** the else block statements are executed



Illustration: If-else Statement

[Click to Continue](#)



```
int studentId=900;  
int studentAge=18;  
int marks=80;  
If((studentAge>20) && (marks >70)){  
    System.out.println("Eligible for graduation");  
}  
Else  
{  
  
    System.out.println("Not Eligible for graduation");  
}  
}
```

If both the condition age and marks are satisfied this message will be printed.

If the conditions in if block is not satisfied, this block will be fired.



If-else if Statement

Click to Continue



This is an extension of if else where the else if block has a condition to check. If not met the flow goes to the `else block

We can have any number of else-if block but only one block will be executed.

Syntax:

```
if( <condition1> ) {  
    statements  
}  
else if(<condition2>){  
    statements  
}  
else{  
    statements  
}
```

If condition1 is **satisfied** these statements are executed.

If condition 1 is **not satisfied** and **condition2 is satisfied** these statements are executed.

If both conditions **are not satisfied** these statements are executed.



Illustration multiple else if Statement

Click to Continue



```
package com.statement.demo;  
  
public class IfElseIfExample {  
  
    public void whichQuarter(int month) {  
        if (month > 0 && month <= 3) {  
            System.out.println("Quarter 1");  
        } else if (month >= 4 && month <= 6) {  
            System.out.println("Quarter 2");  
        } else if (month >= 7 && month <= 9) {  
            System.out.println("Quarter 3");  
        } else {  
            System.out.println("Quarter 4");  
        }  
    }  
}
```

We have two Else If block



What is a nested if statement?

[Click to Continue](#)



The if statement in java can be nested, in other words, an if statement can be present inside another if statement

Example:

The discount % of T.V is calculated based on the below criteria

If T.V is LED, check the screen size

If screen size is 32, discount % = 10

If screen size is 46, discount % = 15

If T.V is LCD, discount % = 5



Nested IF Implementation

[Click to Continue](#)



```
public void checkDiscount() {  
  
    String typeOfTV = "LED";  
    int sizeofTV = 32;  
    int discount;  
  
    if ("LED".equals(typeOfTV)) {  
        if (sizeofTV == 32) {  
            discount = 10;  
        } else if (sizeofTV == 46) {  
            discount = 15;  
        }  
    } else if ("LCD".equals(typeOfTV)) {  
        discount = 5;  
    }  
}
```

If the type of TV is LED, then size check is done.
Here size check if statement is the nested if.

