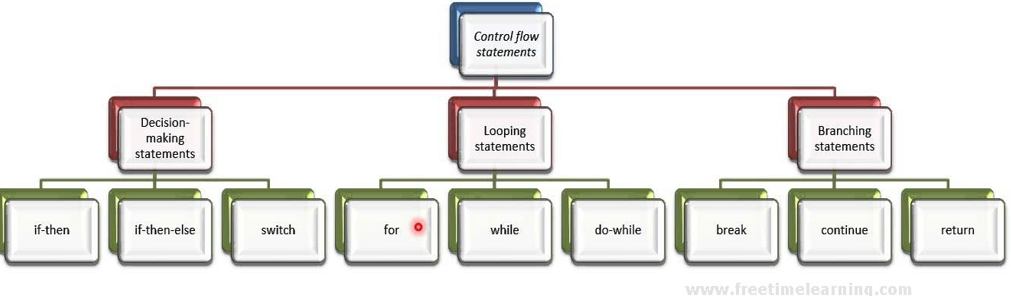
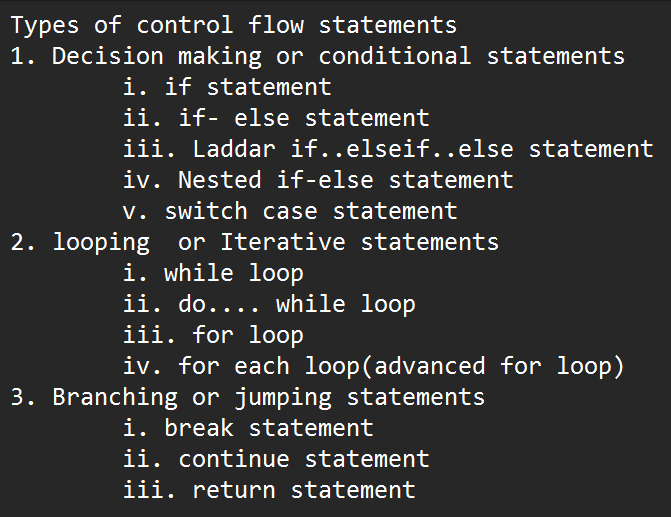
# **Java Notes**

1. Control Statements





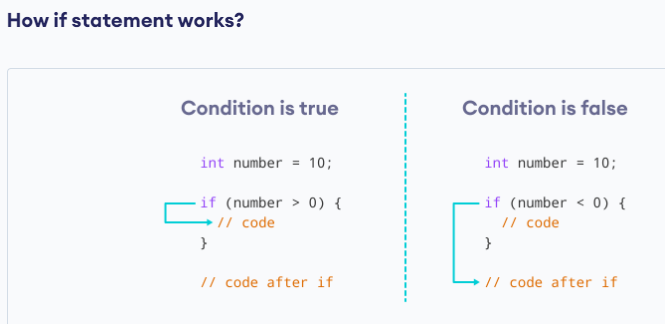
Types of Decision making or conditional statements

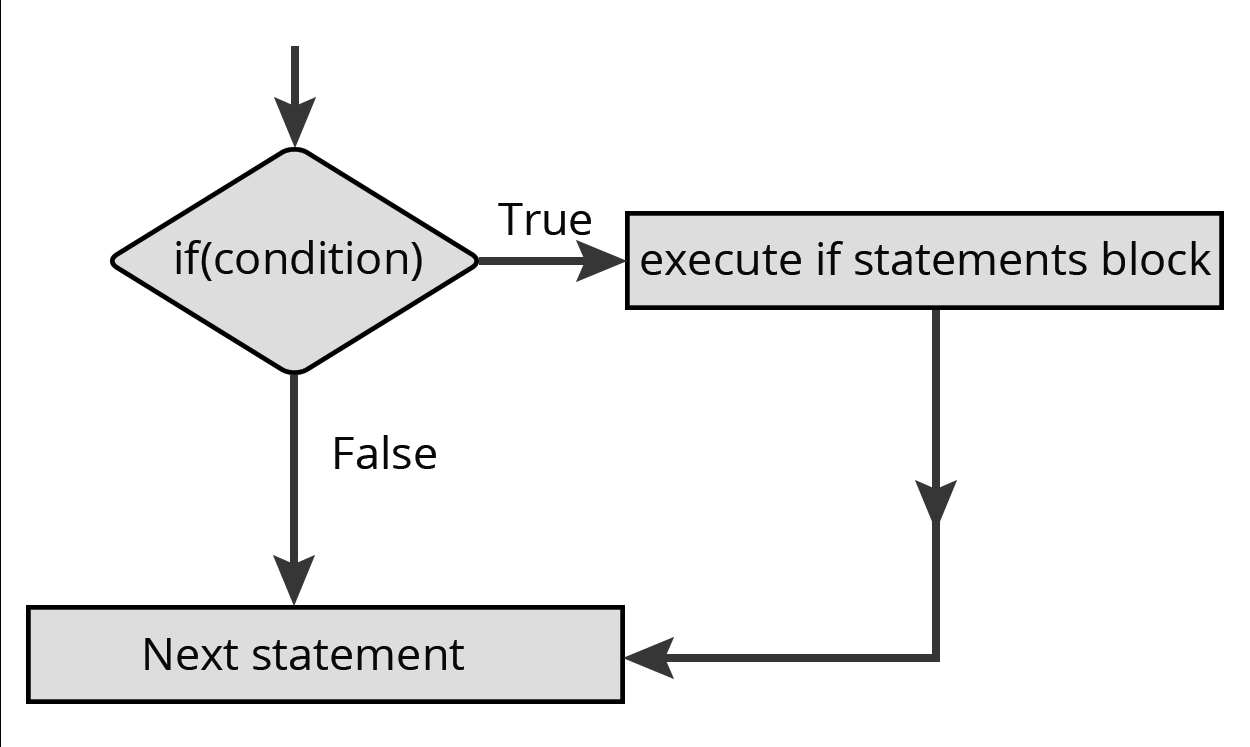
* 1. Simple if statement

***Syntax :***

*If(condition)  
    {*

*statement1;  
    }*





***Code :***

class PrintPassFail  
{  
  public static void main(String arg[])  
    {  
   int marks = 62;  
        if(marks > 35)

        {  
            System.out.println("Pass");

         }

System.out.println(“program end”);

   }

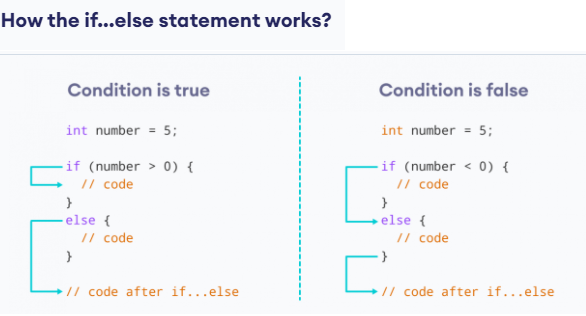
}

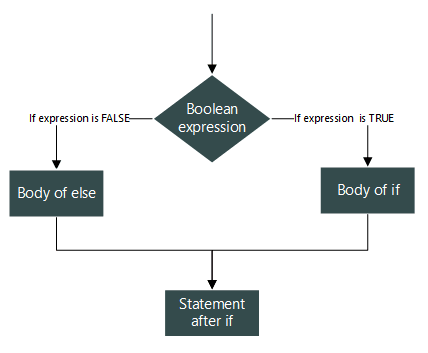
* 1. If else conditional statement

Either if block or else block will be executed.

***Syntax :***

If(condition)  
    {  
        statement1;  
    }  
    else  
    {  
        statement2;  
    }





class PrintPassFail  
{  
    public static void main(String arg[])  
    {  
        int marks = 32;  
        if(marks > 35)             
            System.out.println("Pass");             
        else  
         {  
            System.out.println("Fail");              
     System.out.println(“Better luck next time);   
   }

      }  
}

Unary operator 🡪 ++, -- 🡪 only one operand 🡪 i++, b-- 🡪 i, b are operands

Binary operator 🡪 all arithemetic operators are binary operators🡪 two operands

A+B 🡪 A, B🡪 operands

8.2.1Ternary Operator or conditional operator

The conditional operator (? :) is a ternary operator that takes three operands. It works similar to if else statement.

2+3-6\*5 🡪 arithmetic expression 🡪 arithmetic values

A= 2+3-6\*5 🡪 arithmetic statement

5>18 🡪 Boolean expression 🡪 false

5<=18 🡪 Boolean expression 🡪 true

**Syntax:**

Boolean expression? Value1:value2;

Condition?value1:value2;

Value1, value 2 may be any constants (integer or float or char or string), or condition or expression or statament

Boolean expression means relational expression or logical expression 🡪 always will return Boolean value

**Example:**

char grade = mark > 90 ? 'A' : 'B';

String result = mark>=50? “pass”:”faill”;

package javabasics;

import java.util.Scanner;

public class TernaryOperator {

public static void main(String[] args) {

Scanner ip = new Scanner(System.***in***);

System.***out***.println("Enter a mark;");

int mark = ip.nextInt();

//Relational expression in Ternary operator

String result= mark>=50?"you passed in this subject":"you fail in this subject";

System.***out***.println(result);

//nested ternary operator

int a=200, b=50, c=100;

int biggest= (a>b?(a>c?a:b):(b>c?b:c));

System.***out***.println("Biggest number is "+biggest);

//logical expression in Ternary operator

int age=4;

String citizen = "indian";

String e = ((age>=18) && (citizen=="indian"))?"eligible to vote in india":"not eligile to vote in india";

System.***out***.println(e);

}

}

Output:

Enter a mark;

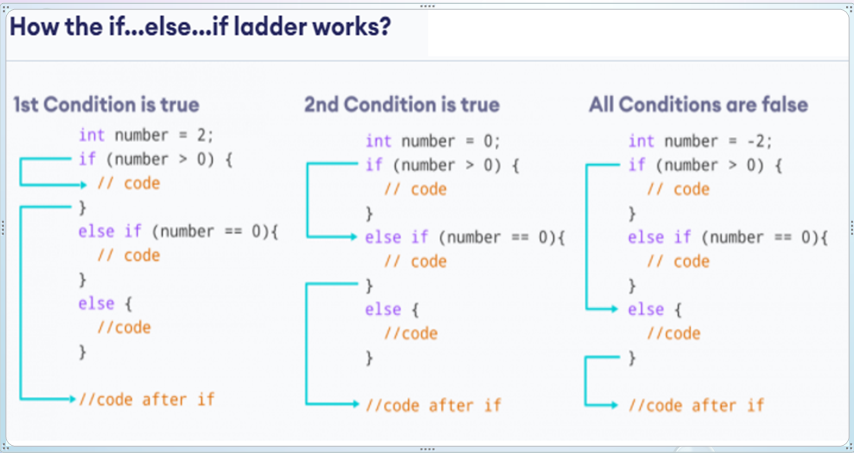
77

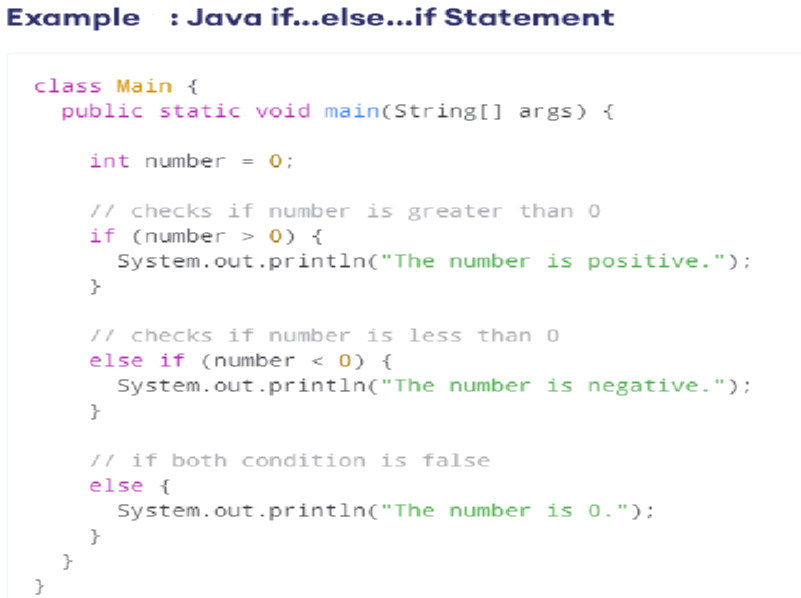
you passed in this subject

Biggest number is 200

not eligile to vote in india

* 1. Ladder if else statement
* **If.. else...if** ladder is used to execute one block of code, when many block of codes are available.
* Out of many conditions, one condition will be true, that corresponding block will be executed.





public class LaddarElseIf // main class

{

public static void main(String args[])

{

System.out.println("welcome to java programming");

int a=200, b=167,c=125;

// Laddar elseif

if ((a>b) && (a>c))

System.out.println("Greatest number is "+a);

else if (b>c)

System.out.println("Greatest number is "+b);

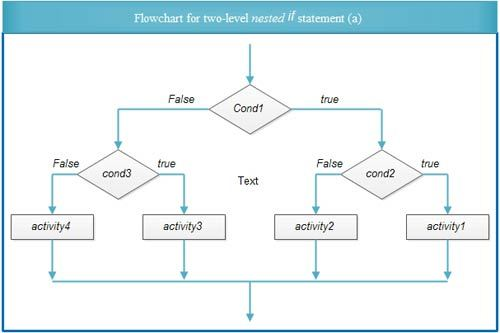
else

System.out.println("Greatest number is "+a);

}

}

* 1. Nested if else conditional statement
* In nested if..else statement, we can have if..else inside another if...else statement.



public class NestedIf // main class

{

public static void main(String args[])

{

System.out.println("welcome to java programming");

int a=200, b=167,c=125;

if(a>b)

{

if(a>c)

System.out.println("Greatest number is "+a);

else

System.out.println("Greatest number is "+c);

}

else

{

if(b>c)

System.out.println("Greatest number is "+b);

else

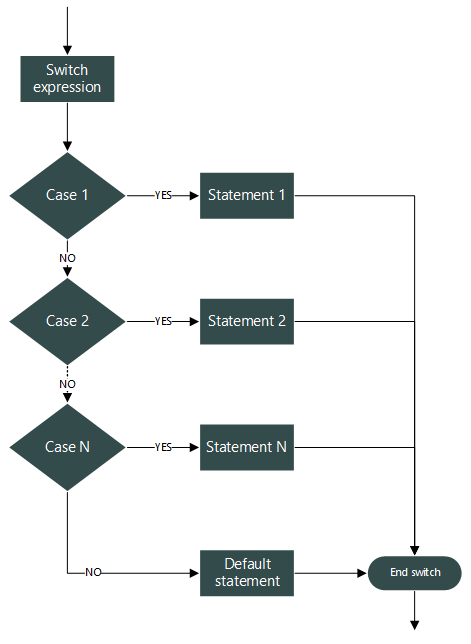
System.out.println("Greatest number is "+c);

}

}

}

* 1. **Switch statements**
* Similar to ladder if…else
* Break statement is used to come out of switch block
* If break is removed in any of the cases, the remaining cases, it takes as normal statement, executes sequentially.



Syntax

switch(expression)  
    {  
        case value1:  
             // statement1;  
             break;  
        case value2:  
             // statement2;  
            break;

..........  
        case value n:  
             // statement n;  
            break;

        default:  
            // statement3;  
            break;  
    }

Sample Program:

class PrintDirection  
{  
    public static void main(String arg[])  
    {  
        char direction = 'S';  
        switch( direction )  
        {  
            case 'E':   
                System.out.println("East"); // LINE A  
                break;  
            case 'W':   
                System.out.println("West"); // LINE B  
                break;  
            case 'S':   
                System.out.println("South"); // LINE C  
                break;  
            case 'N':   
                System.out.println("North"); // LINE D  
                break;  
            default:   
                System.out.println("Unknown Direction"); // LINE E  
                break;  
        }

}

}