

JDK JRE JVM

Date _____
Page _____

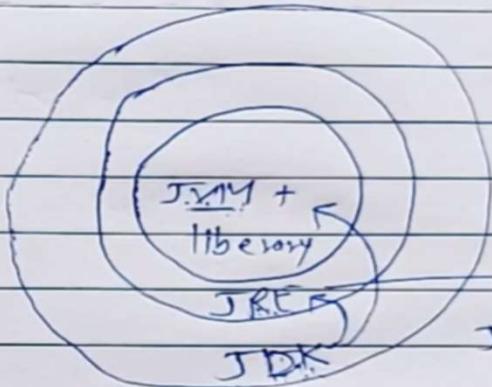
JDK :- JDK Stands for Java development kit, it internally contains JRE & JVM where JRE Stands for Java Runtime Environment AND JVM Stands for Java Virtual machine.
JDK provides all the tools to work with Java language.

JRE :- JRE Stands for Java runtime environment, it's provide an environment to execute the Java program. It internally contains JVM which is responsible to execute a Java program.

JVM :- JVM Stands for Java Virtual machine, it is the software in the form of interpreter written in 'C' language through which we can execute our Java programs.

JDK JRE JVM

Page _____



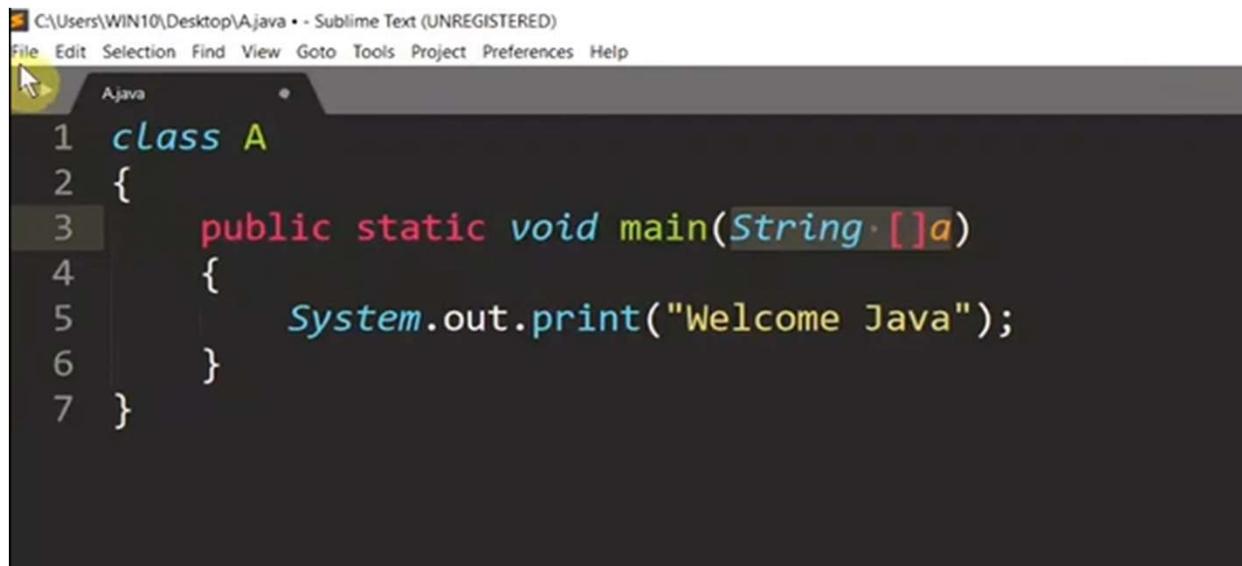
$$\text{JRE} = \text{JVM} + \text{library}$$
$$\text{JDK} = \text{JRE} + \text{JVM}(\text{library})$$

Java

Date _____
Page _____

What is Java? With Syntax & Example.

Java is a class based high level object oriented programming language developed by "James Gosling" and his friends in the year 1991.



A screenshot of Sublime Text showing a Java file named A.java. The code defines a class A with a main method that prints "Welcome Java".

```
1 class A
2 {
3     public static void main(String []a)
4     {
5         System.out.print("Welcome Java");
6     }
7 }
```

Java History

Java is a totally computer-based programming language developed by Sun microsystems (James Gosling, Mike Sheridan & patric naughton).

In the year 1991 James Gosling & his friend start a project.
Team → Green team

Features of Java

Page

Object Oriented

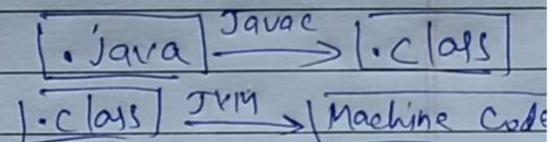
Platform independent

Simple language

Secure

Portable

Compiled & Interpreted



Robust

Distributed

Java Comments V.V.I

Date : / /

What is Comment? full explanation.

Comments are the statements that are totally ignored by the compiler & interpreter.

Why we use Comment:-

- I) The Comments can be used to provide explanation about the program.
- II) Comments makes program more understandable and readable to the others.

Types :- I) Single line Comment //

II) Multi line Comment /*

III) Documentation Comment

Use of Comments

```
class Test
{
    int a=10;      // Instance Variable
    static int b=20;      // static variable

    public static void main(String[] args) {

        int c=30;      // Local Variable
        final int D=40; // final variable

        Test t=new Test(); // object creation

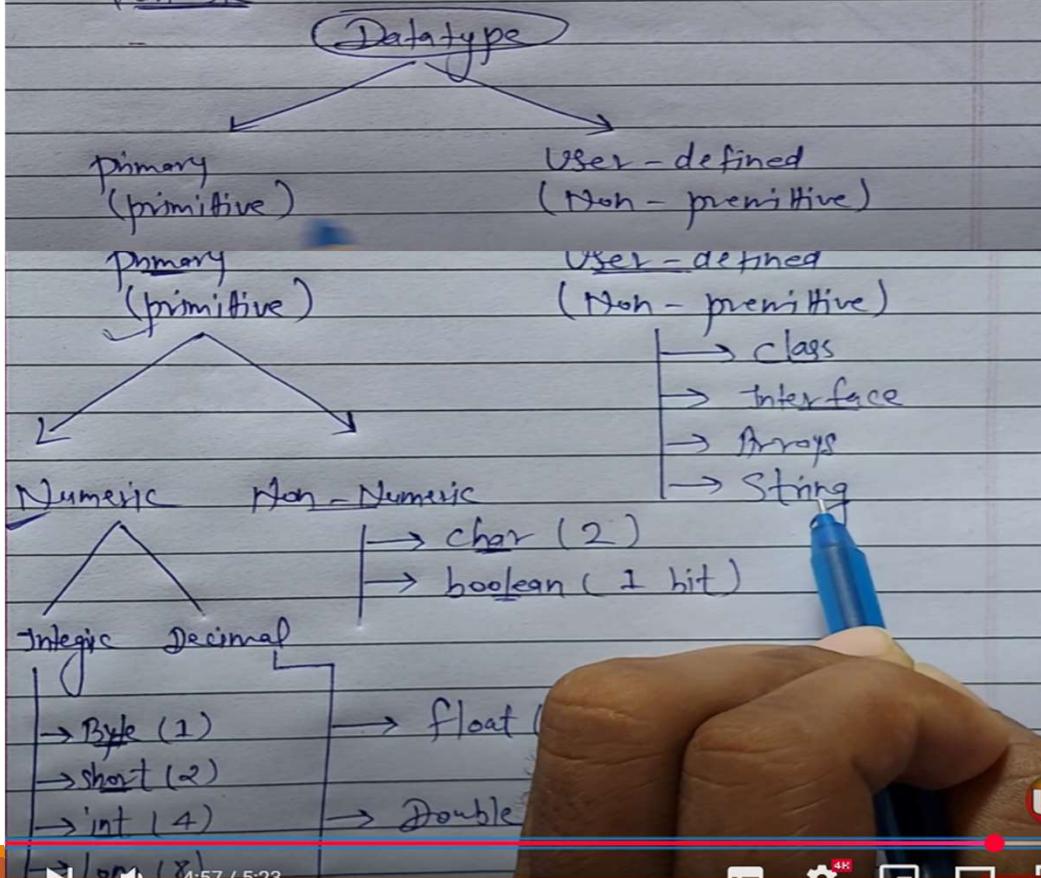
        System.out.println(t.a);
        System.out.println(b);

        System.out.println(c);
        System.out.println(D);
```

Datatype in JAVA

Page

Datatype Specify the different sizes of values that can be stored in the variable.



Variable in Java

Date _____
Page _____

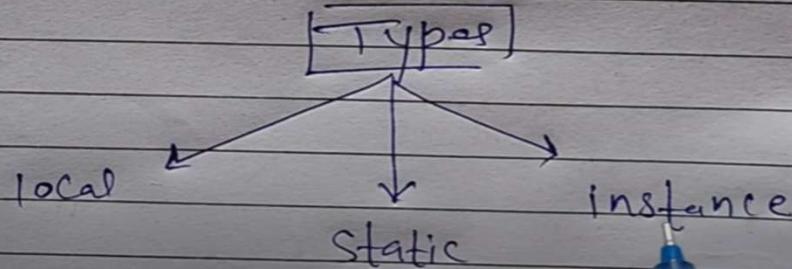
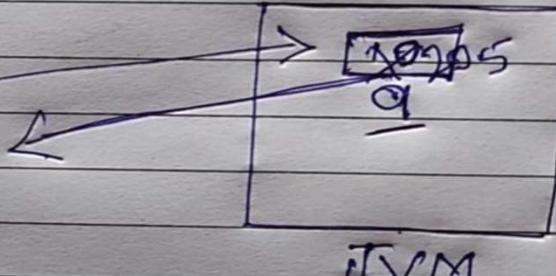
What is Variable? Discuss with types.

Variable is the name of memory location. In other word we can say it is user defined name which is given by user. Variable can store any type of values.

`int a = 10;`

`a = a + 10;`

`float a = 20.5;`

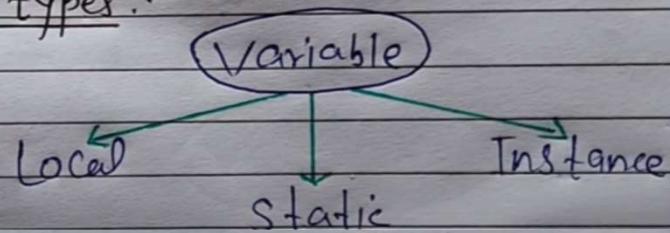


Java variable

Date _____
Page _____

How many types of variable in Java?

Three types:-



Local Variable:- A variable which is declared inside the body of the method or method parameter called local variable.

Syntax:- void fun(int a)
 {
 int x; // local variable
 }

Instance Variable:- A variable which is declared inside the class but outside of all the methods called instance variable.

Syntax:- class A

[int a;] // instance
p.s.v.m()
↓

Static Variable:- A variable which is declared with the help of static keyword called static variable.

Syntax:- [static int x;]

```
class A
{
    static int b=20;//Static
    int c=30;//Instance
    public static void main(String[] args)
    {
        int a=10;//Local
        A ref=new A();
        System.out.println(a);
        System.out.println(A.b);
        System.out.println(ref.c);
    }
}
```

Static variable without creating an object of the class can be run with the **class. variable** ke naam se

```
class A
{
    static int a=10;
    void fun()
    {
        int b=10;
        System.out.println(a+" "+b); //11 10
        ++a; ++b; //11 11
    }
    public static void main(String[] args)
    {
        A r=new A();
        r.fun();
        r.fun();
    }
}
```

Static variable ki
life time validity
hoti he

Java (INPUT & OUTPUT)

INPUT (Scanner class) :-

Scanner is a pre-defined class in java which is available in `java.util` package. It is used to get user input.

Rule ① :- If we use Scanner class, must have to create object of Scanner class.

Syntax:-

`Scanner object_name = new Scanner(System.in);`

② Scanner class methods:-

- i) `nextLine();` // String
- ii) `nextInt();` // Integer
- iii) `nextFloat();` // Floating
- iv) `nextBoolean();` // true or false
- v) `nextDouble();` // double

③ import Scanner class package at the top line of program:-

Syntax:- `import java.util;`

OUTPUT :- (`System.out.print()`)

It's an output statement in Java through which we can print the variables, expressions and many more content.

System:- ① pre defined class.
② java.lang package.

A screenshot of a Java code editor showing a simple program named 'A.java'. The code imports the Scanner class from java.util and defines a class A with a main method. Inside the main method, it prints a prompt, creates a Scanner object to read integer input, reads the input into variable a, and then prints the value of a. An orange arrow points from the handwritten note 'Integer input' to the line `a=obj.nextInt();`.

```
1 import java.util.Scanner;
2 class A
3 {
4     public static void main(String args[])
5     {
6         int a;
7         System.out.print("Enter Data ");
8         Scanner obj=new Scanner(System.in);
9         a=obj.nextInt();
10        System.out.print("Get Data "+a);
11    }
12 }
13 }
```

```
A.java
import java.util.Scanner;
class A
{
    public static void main(String args[])
    {
        String a; |
        System.out.print("Enter Data ");
        Scanner obj=new Scanner(System.in);
        a=obj.nextLine();
        System.out.print("Get Data "+a);
    }
}
```

String input

```
userInput.java
import java.util.Scanner;
class userInput
{
    public static void main(String args[])
    {
        int a,b;
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter value ");
        a=obj.nextInt();
        b=obj.nextInt();
        System.out.print(a+" "+|+b);
    }
}
```

I

Java Token

Date : / /

What is Tokens? full explanation.

- > Token is the smallest elements of a program that is identified by a compiler.
Every Java statements and expressions are created using tokens.

Example:- Class A

```
class A {  
    public static void main(String args[]) {  
        int a=10, b=20;  
        int c=a+b;  
        System.out.println(c);  
    }  
}
```

Tokens :-

- i) Keywords
- ii) Identifiers
- iii) Operators
- iv) Separators
- v) Literals

Java Keyword

Date : / /

i

What is keyword? full explanation.

- Keyword's are the reserved word whose meaning is already defined in the Compiler Called Keywords.

Java Keyword's :-

| | | | | |
|-------------------|------------|------------|--------------|----------|
| int | Continue | instanceof | Super | boolean |
| Short | break | interface | this | abstract |
| byte | default | native | switch | base |
| long | double | new | synchronized | Catch |
| if | extends | package | throw | class |
| else | implements | private | throws | Const* |
| for | final | protected | try | goto* |
| while | finally | public | transient | Strictfp |
| do for | float | return | void | enum** |
| char | import | Static | volatile | assert* |

Java (operator)

Page

What is Operators? With Types.

Operator is a symbol that is used to perform operations according to user requirement. (variable & values)

- Types:-
- ① Arithmetic (+, -, *, /, %)
 - ② Relational (<, >, >=, <=, !=, ==)
 - ③ Logical (&, ||, !)
 - ④ Inc/Dec ← pre inc / post inc (++exp, ++exp++)
 pre dec / post dec (--exp, exp--)
 - ⑤ Assignment → Simple (=)
 → Compound (+=, -=, *= etc)
 - ⑥ Ternary (?:)
 - ⑦ Bitwise (AND, OR, NOT, complement)



Arithmetic Operators

```
/* Arithmetic Operator Example...! */
import java.util.Scanner;
class Arithmetic
{
    public static void main(String args[])
    {
        int a,b;
        System.out.print("Enter two Numbers... ");
        Scanner obj=new Scanner(System.in);
        a=obj.nextInt();
        b=obj.nextInt();
        System.out.println("Addition "+(a+b));
        System.out.println("Subtraction "+(a-b));
        System.out.println("Multiplication "+(a*b));
        System.out.println("Division "+(a/b));
        System.out.print("Remainder "+(a%b));
    }
}
```

Relational Operators

```
/* Relational Operator Example...! */  
import java.util.Scanner;  
class Relational  
{  
    public static void main(String[] args) {  
        int a,b;  
        System.out.print("Enter two Numbers.. ");  
        Scanner obj=new Scanner(System.in);  
        a=obj.nextInt(); //Greater  
        b=obj.nextInt(); //Lesser  
        System.out.println("true/false "+(a<b));  
        System.out.println("true/false "+(a>b));  
        System.out.println("true/false "+(a<=b));  
        System.out.println("true/false "+(a>=b));  
        System.out.println("true/false "+(a==b));  
        System.out.print("true/false "+(a!=b));  
    }  
}
```

Logical Operators

```
/* Logical Operator Example...! */

class Logical
{
    public static void main(String[] args) {
        System.out.println("Logical AND ");
        System.out.println((10>5) && (2>1));      //true
        System.out.println((10>5) && (2<1));      //false
        System.out.println((10<5) && (2<1));      //false

        System.out.println("Logical OR ");
        System.out.println((10>5) || (2>1));      //true
        System.out.println((10>5) || (2<1));      //true
        System.out.println((10<5) || (2<1));      //false

        System.out.println("Logical NOT ");
        System.out.println(!(10>5));      //false
        System.out.println(!(10<5));      //true
    }
}
```

Logical Operators

```
/* Incr/Decr Operator Example...! */
```

```
class PrePost
```

```
{
```

```
    public static void main(String[] args) {
```

```
        int a=10;
```

```
        System.out.println(a);
```

```
        System.out.println("PrePost Incr.. ");
```

```
        System.out.println(a++);//10 (11)
```

```
        System.out.println(++a);//12
```

Post increment

Pre increment

Post Decrement

Post Decrement

```
}
```

```
}
```

Assignment.java

```
/* Assignment Operator Example...! */

class Assignment
{
    public static void main(String[] args) {

        int a;
        a=10; //Simple
        System.out.println(a);

        a+=10; //Compound (a=a+10)
        System.out.println(a);

        a-=10; //Compound (a=a-10)
        System.out.println(a);

    }
}
```

```
/* Ternary Operator Example...! */
class Ternary
{
    public static void main(String[] args) {

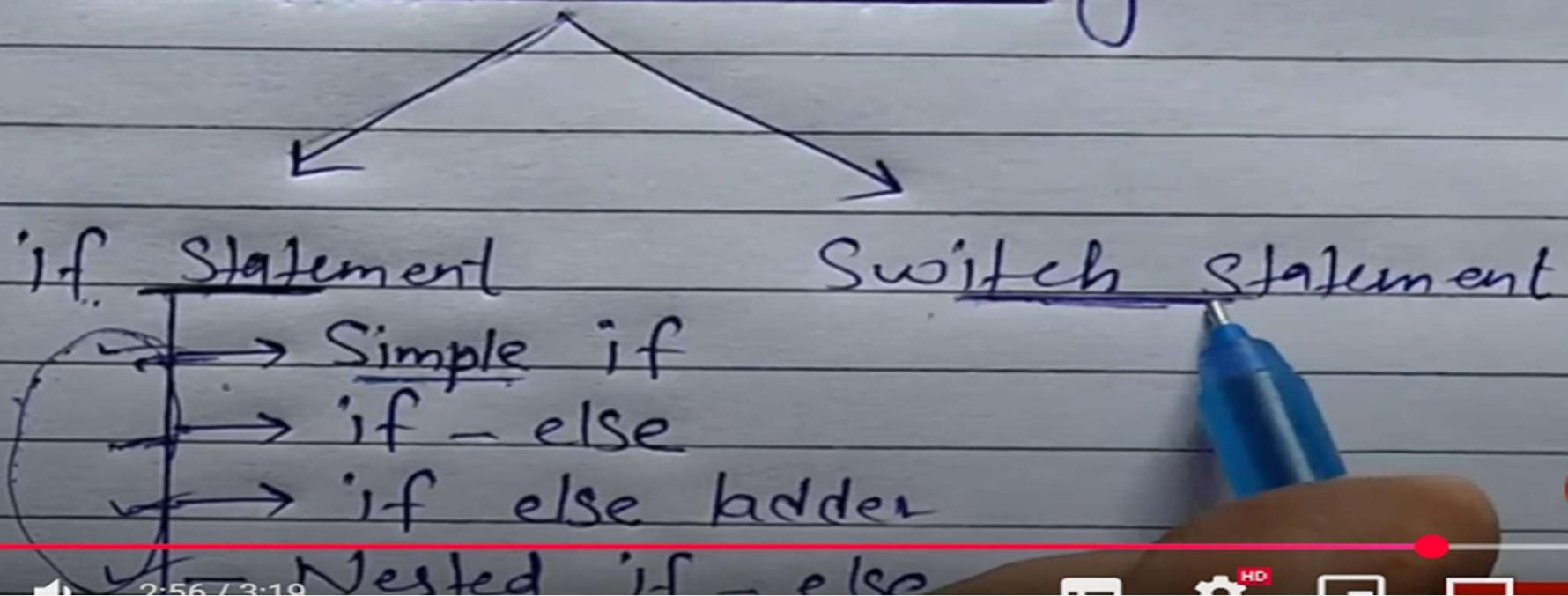
        int a=10,b=20,Max;
        Max=(a>b) ? a : b;
        System.out.print(Max);
    }
}
```

```
ternary.java
/* Ternary Operator Example...! */
class Ternary
{
    public static void main(String[] args) {

        int a=10,b=4,c=2,Max;
        Max=(a>b) ? (a>c ? a : c) : (b>c ? b : c);
        System.out.print(Max);
    }
}
```

Java

Decision making

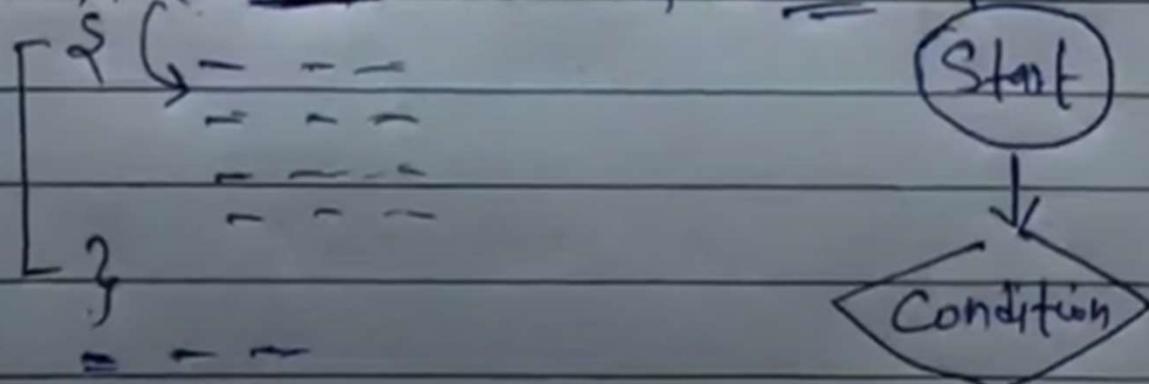


if- Statement

Page No.

Simple if- Statement :- It is used when we want to test a Condition.

Syntax:- if (Condition) flowchart:



```
/* Simple if Statement...! */
import java.util.Scanner;
class Simple
{
    public static void main(String[] args) {

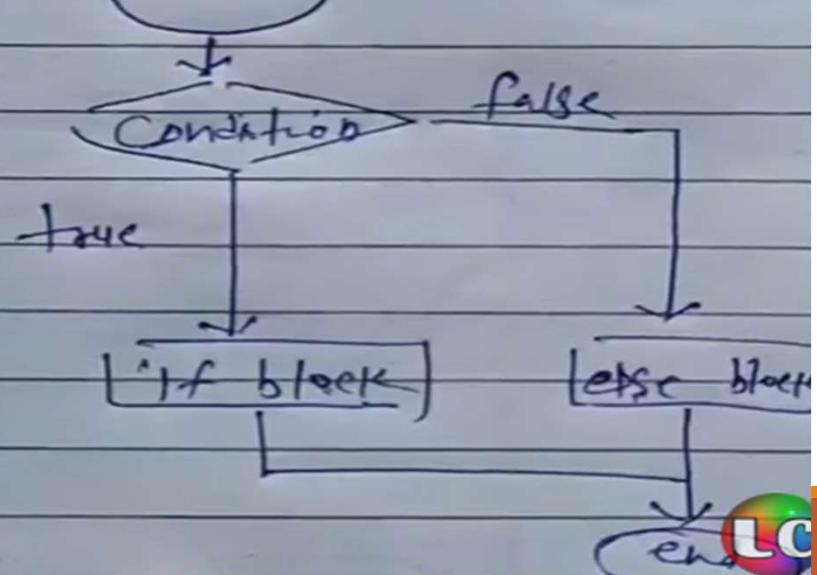
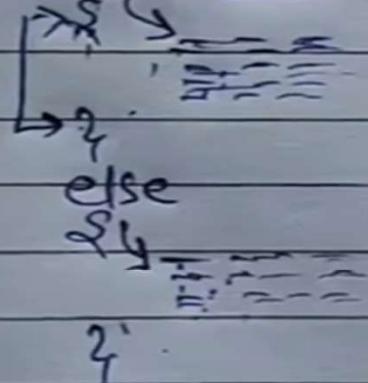
        int age;
        System.out.println("Enter Your Age..!");
        Scanner r=new Scanner(System.in);
        age=r.nextInt();
        if(age>=18)
        {
            System.out.println("Eligible for votes..!");
        }
        System.out.print("Thank You..!");
    }
}
```

2. if-else Statement

Page No.

It is used to execute two statements either if statement or else statement for a single condition.

Syntax - if (condition) { } Flowchart :-



```
/* if-else Statement...! */
import java.util.Scanner;
class ifElse
{
    public static void main(String[] args) {

        int n;
        System.out.println("Enter any Numbers..! ");
        Scanner r=new Scanner(System.in);
        n=r.nextInt();
        if(n>=0)
        {
            System.out.println("+ve Number");
        }
        else
        {
            System.out.println("-ve Number");
        }
    }
}
```

else if ladder Statement

Page No.:

It is used when we have only one if block, multiple else-if blocks and at last one else block.

Syntax:- if (condition)

X { = = =
 } = = =

 else if (condition)

 { = = =
 } = = =

 else if (condition)

 { = = =
 } = = =

 else

 { = = =
 } = = =

```
elseif.java
/*
 *      if-else if Statement...!      */
import java.util.Scanner;
class elseif
{
    public static void main(String[] args) {

        int marks;
        System.out.println("Enter marks ");
        Scanner ref=new Scanner(System.in);
        marks=ref.nextInt();

        if(marks>80)
        {
            System.out.println("Topper");
        }
        else if(marks<80 && marks>=60)
        {
            System.out.println("First");
        }
        else
        {
            System.out.println("Second ");
        }
    }
}
```

Java

| | | | |
|----------|--|--|--|
| Date | | | |
| Page No. | | | |

4. Nested if else Statement

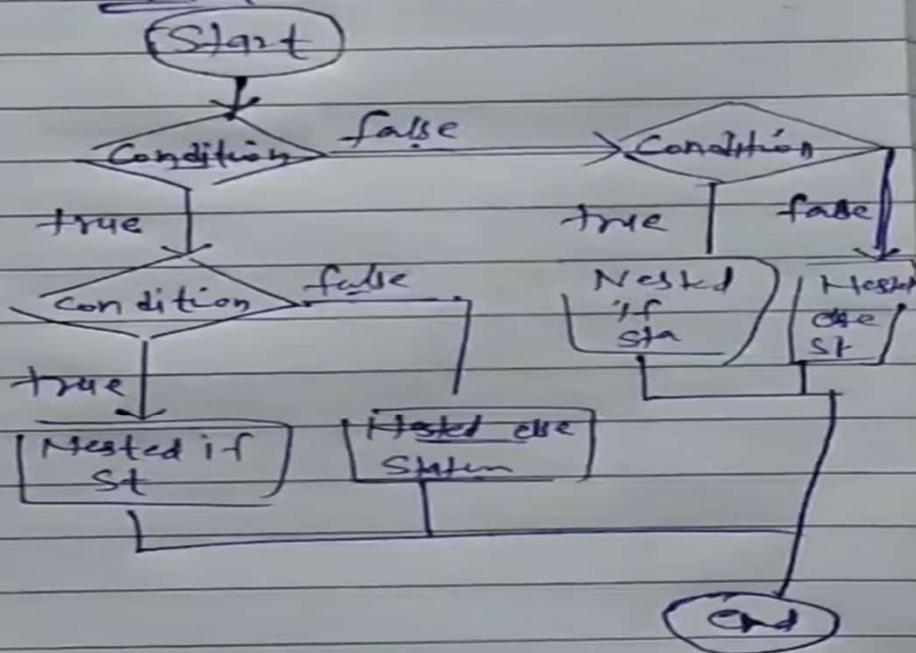
Whenever we define if else block, inside this if else block we define another if else block is called Nested if else statement.

Syntax:- if (Condition)

```
{  
    if (Condition)  
    {  
        ...  
    }  
    else  
    {  
        ...  
    }  
}
```

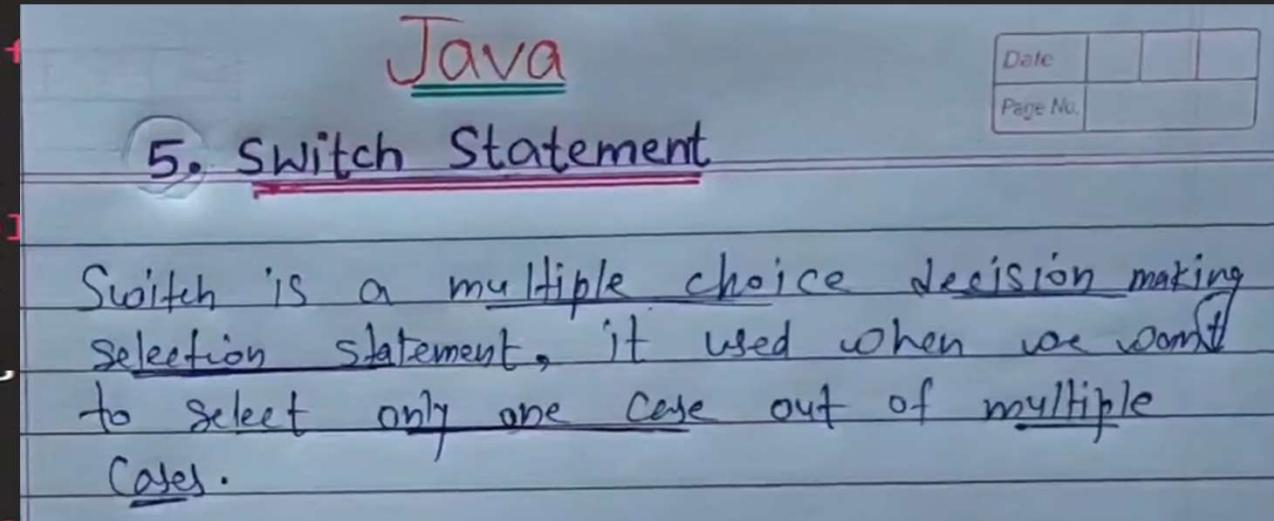
```
{  
    if (Condition)  
    {  
        ...  
    }  
    else  
    {  
        ...  
    }  
}
```

Flowchart:-



```
/* Nested if else Statement */
class Nested
{
    public static void main(String[] args) {

        int a=10,b=20,c=30;
        if(a>b)
        {
            if()
            {
                System.out.println(b);
            }
            else
            {
                System.out.println(a);
            }
        }
    }
}
```



```
if(b>c)
{
    System.out.println(b);
}
else
{
    System.out.println(a);
}
```

Java

5. Switch Statement

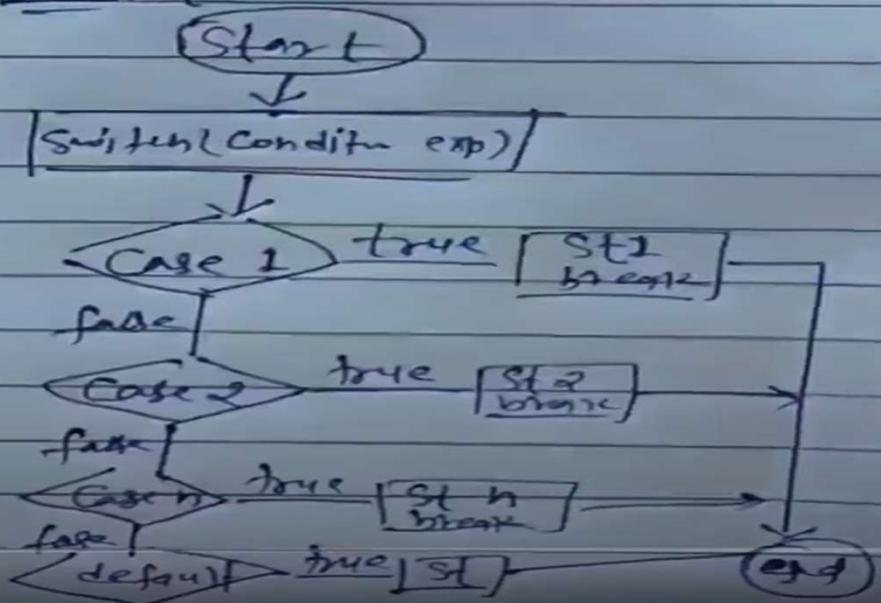
| | | | |
|----------|--|--|--|
| Date | | | |
| Page No. | | | |

Switch is a multiple choice decision making selection statement, it is used when we want to select only one case out of multiple cases.

Syntax :- `switch(exp)`

```
  {  
    Case 1: Statement 1;  
    break;  
    Case 2: St 2;  
    break;  
    ...  
    Case n: St n;  
    break;  
    default: St ;  
  }
```

Flowchart :-



```
Switch.java x
        /* Switch Statement */
class Switch
{
    public static void main(String[] args) {

        int a=10,b=20,ch;
        System.out.print("Enter User Choice..!\\n");
        Scanner r=new Scanner(System.in);
        ch=r.nextInt();
        switch(ch)
        {
            case 1:System.out.print("Sum " + (a+b));
            break;
            case 2:System.out.print("Sub " + (a-b));
            break;
            case 3:System.out.print("Multi " + (a*b));
            break;
            case 4:System.out.print("Div " + (a/b));
            break;
            default:System.out.print("Invalid Choice...! ");
        }
    }
}
```

Java

| | |
|----------|--|
| Date | |
| Page No. | |

Looping Statement

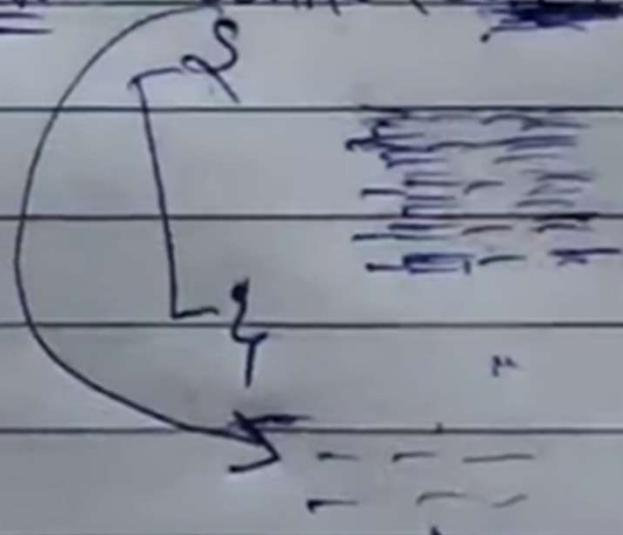
Whenever we have to repeat certain Statement several times is called loop.

1. While loop (Java)

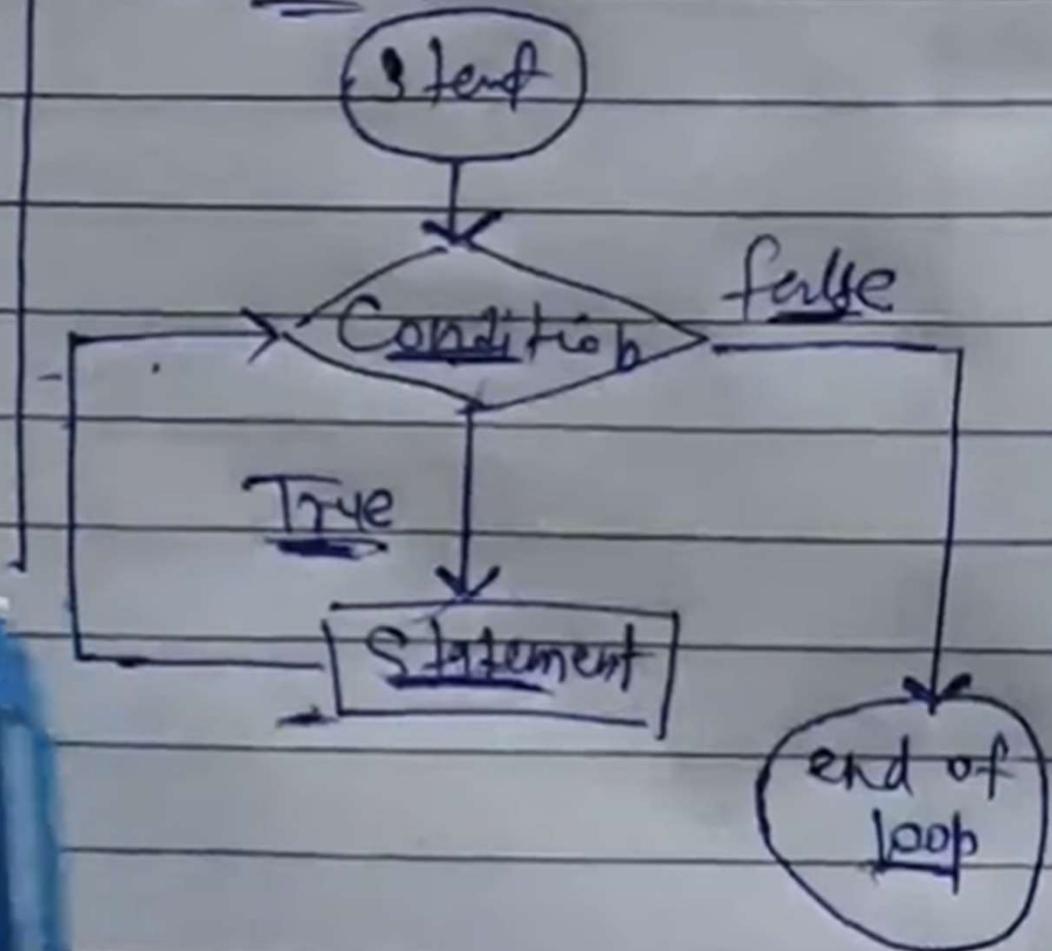
while loop is a pre-test loop, it is used when we don't know the no. of iterations in advance.

It is also known as entry control loop.

Syntax:- while (condition)



Flowchart:-



```
/* While Loop Example...! */
import java.util.Scanner;
class While
{
    public static void main(String[] args) {

        int n;
        System.out.print("Enter value for Condition \n");
        Scanner ref=new Scanner(System.in);
        n=ref.nextInt();

        while(n>=0)
        {
            System.out.print("Learn Coding ");
        }
    }
}
```

```
1          /* While Loop Example...! */
2 import java.util.Scanner;
3 class While
4 {
5     public static void main(String[] args) {
6
7         int n=1;
8         while(n<=10)
9         {
10             System.out.print("Learn Coding\n ");
11             ++n;
12         }
13     }
14 }
```

Create new
line

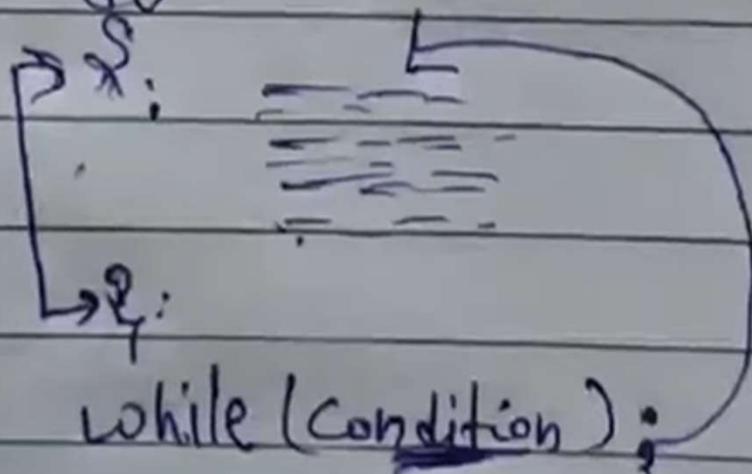
| | |
|-----------|--|
| Px, + N9. | |
|-----------|--|

2. do-while loop (Java)

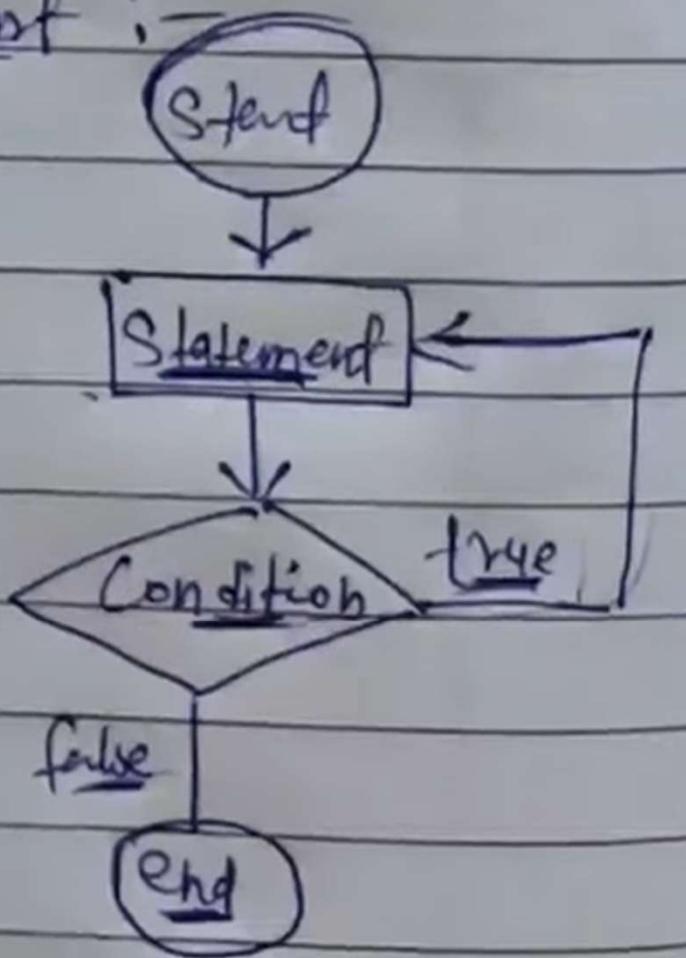
Do-while loop is a post-test loop, it is used when we want to execute loop body at least once even condition is false.

It is also known as exit control loop.

Syntax:- do



Flowchart :-



DoWhile.java

```
/* do-while Loop Statement...! */

class DoWhile
{
    public static void main(String[] args) {

        int n=1;
        do
        {
            System.out.print(n+" ");
            ++n;
        }
        while(n<=10);
    }
}
```

| | | |
|----------|--|--|
| Date | | |
| Page No. | | |

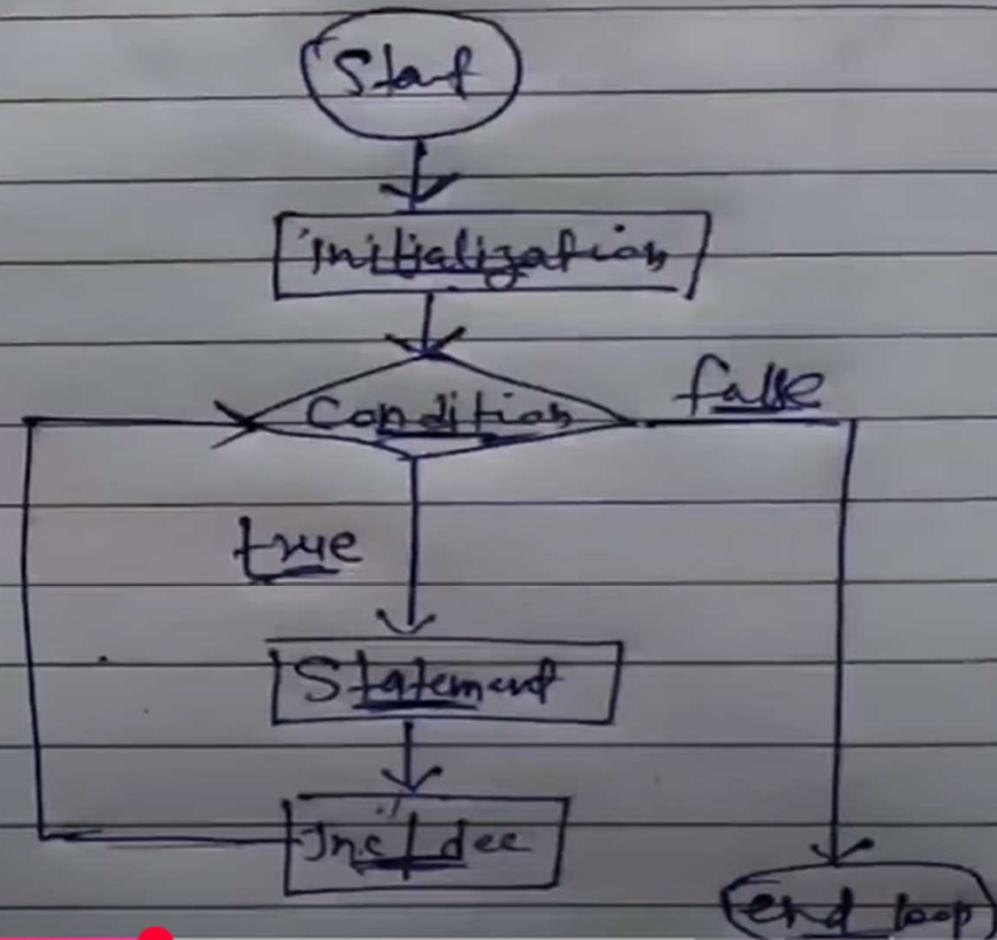
3. for loop (Java)

for loop is the most commonly used loop,
it is used when we want to perform
initialization, condition and increment operation
in single line.

Syntax:-

```
for (initialization; Condition; in/dec)  
{  
    Statement  
}  
}
```

Flowchart :-

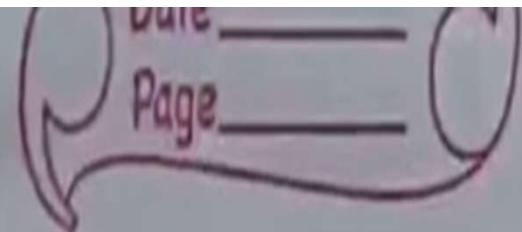


```
public static void main(String[] args) {  
    for (int i=1; i<=10 ; i++ )  
    {  
        System.out.print(i + " ");  
    }  
}
```

For.java

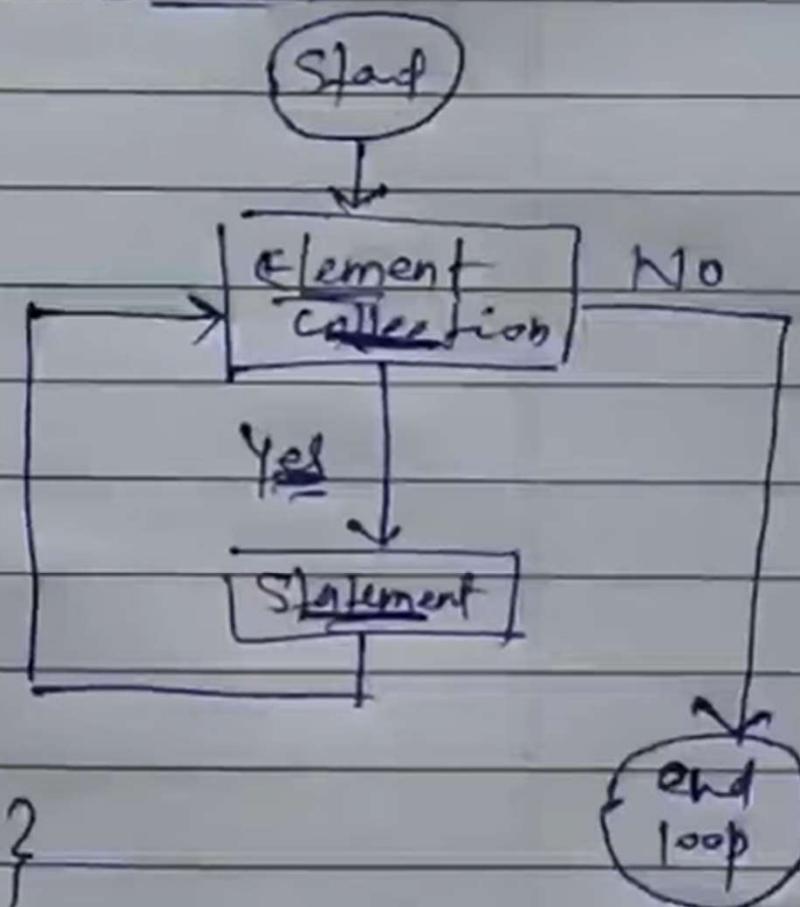
```
/* for Loop Example...! */  
class For  
{  
    public static void main(String[] args) {  
        for (int i=10; i>=1 ; i-- )  
        {  
            System.out.print(i + " ");  
        }  
    }  
}
```

4. for each loop (Java)



for each loop mainly used to fetch the values from a collection like array,

Syntax for (data-type Var1 : Var2) | Flowchart



```
untitled
/*
 *  for-each Loop Statement..!
 */
class ForEach
{
    public static void main(String[] args) {
        int a[]={10,20,30,40,50};

        for ( int b : a ) {
            System.out.print(b+ " ");
        }
    }
}
```

```
/* for-each Loop Statement..! */

class ForEach
{
    public static void main(String[] args) {

        int a[]={10,20,30,40,50};

        for ( int b : a ) {
            System.out.print(b + " ");
        }
    }

    for (int i=0;i<a.length ;i++ ) {
        System.out.print(a[i]+ " ");
    }
}
```

For loop

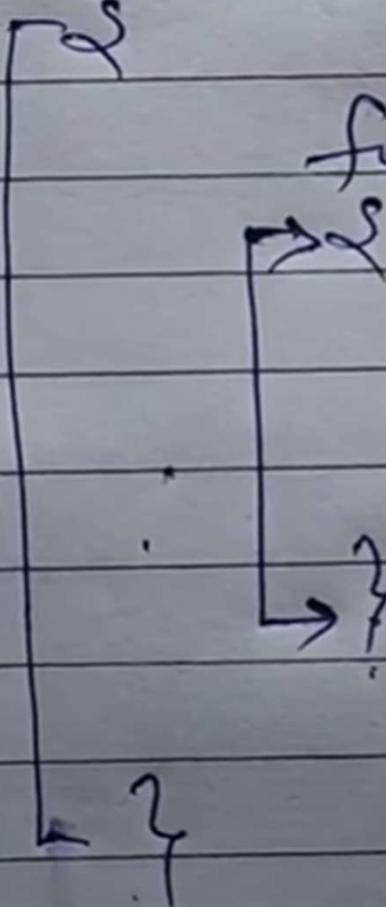
Java

| | | |
|----------|--|--|
| Date | | |
| Page No. | | |

Nested for loop

A for loop which contain inside another for loop is called nested for loop.

Syntax:- `for (initialization; condition; increment/decr)`



`for (initialization; condition; increment/decr)`

- - -
- - -
- - -

Ex →

| | | | | |
|---|---|---|---|---|
| * | * | * | * | * |
| x | x | x | x | x |
| — | — | — | — | — |
| x | x | x | x | x |
| x | x | x | x | x |

```
/* Nested For Loop Example..! */\nclass Nested\n{\n    public static void main(String[] args) {\n\n        int i,j;\n\n        for (i=1;i<=5 ;i++ ) //row\n        {\n            for (j=1;j<=5 ;j++ ) //column\n            {\n                System.out.print("* ");\n            }\n            System.out.println();\n        }\n    }\n}
```

```

Ex → int i, j;
for(i=1; i<=5; i++) // i=1 & 3 & 5
{
    for(j=1; j<=5; j++) // j=1 & 3 & 5
        cout(" * ");
    cout("\n"); // for New line
}

```

C:\Users\WIN10\

```

* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

```

$i = \begin{cases} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{cases}$

| j = 1 | * | * | * | * |
|-------|---|---|---|---|
| 1 | * | * | * | * |
| 2 | * | * | * | * |
| 3 | * | * | * | * |
| 4 | * | * | * | * |
| 5 | * | * | * | * |

```
        /* Odd Even Program */
import java.util.Scanner;
class OddEven
{
    public static void main(String[] args) {

        int n;
        System.out.print("Enter any Number ");
        Scanner r=new Scanner(System.in);
        n=r.nextInt(); //23

        if(n%2==0)//23%2==0
        {
            System.out.print("Even Number");
        }
        else
        {
            System.out.print("Odd Number");
        }
    }
}
```

```
    /* Sum of Two Numbers Program */  
import java.util.Scanner;  
class Add  
{  
    public static void main(String args[])  
    {  
        int a,b,sum;  
        System.out.print("Enter two Numbers ");  
        Scanner r=new Scanner(System.in);  
        a=r.nextInt(); //45  
        b=r.nextInt(); //76  
  
        sum= a + b; //45+76=121  
        System.out.print("ADDITION " + sum);  
    }  
}
```

```
Natural.java
1          /* Natural Number Program */
2 import java.util.Scanner;
3 class Natural
4 {
5     public static void main(String[] args) {
6
7         int n;
8         System.out.print("Enter No. of Term ");
9         Scanner r=new Scanner(System.in);
10        n=r.nextInt(); //n=3
11
12        for(int i=1;i<=n;i++) //i=4
13        {
14            System.out.print(i + " ");
15        }
16    }
17 }
```

```
untitled
1      /* Sum of first N Natural Numbers */
2 import java.util.Scanner;
3 class Sum
4 {
5     public static void main(String[] args) {
6
7         int n, sum=0;
8         System.out.print("Enter no of Term ");
9         Scanner r=new Scanner(System.in);
10        n=r.nextInt();
11
12        for(int i=1; i<=n; i++)
13        {
14            sum=sum+i;
15        }
16        System.out.print("ADDITION " + sum);
17    }
18 }
```

Dry Run

int n, i, Sum=0;

Scanner r=new Scanner (System.in);

n=r.nextInt(); // n=3

for(i=1; ~~i<=n~~; i++) // ~~i=1, 2, 3, 4~~

{
 i
 3 <= 3}

Sum=Sum+i; // 0+1=1+2=3+3=6

System.out.print(Sum); // 6

Edit Selection Find View Goto Tools Project Preferences Help

```
Odd.java  
1  /* Print Odd Numbers In Given Range */  
2  import java.util.Scanner;  
3  class Odd  
4  {  
5      public static void main(String[] args) {  
6          int n;  
7          System.out.print("Enter No of Term ");  
8          Scanner r=new Scanner(System.in);  
9          n=r.nextInt();  
10           
11         for(int i=1; i<=n; i=i+2)  
12         {  
13             System.out.print(i + " ");  
14         }  
15     }  
16 }  
17 }
```

Dry Run

int n, i;
S.o.p("Enter no. of Term");
Scanner r=new Scanner(System.in);
n=r.nextInt(); // n=10

for (i=1; i<=n; i=i+2) // i=1 3 5 7 9
{
 S.o.p(i); // 1 3 5 7 9



```
Even.java x Date : /  
1  /* Print Even Numbers in Given Range */  
2 import java.util.Scanner;  
3 class Even  
4 {  
5     public static void main(String[] args) {  
6         int n;  
7         System.out.print("Enter No of Term ");  
8         Scanner r=new Scanner(System.in);  
9         n=r.nextInt();  
10          
11        for(int i=0;i<=n;i=i+2)  
12        {  
13            System.out.print(i + " ");  
14        }  
15    }  
16 }
```

Dry Run

int n, i;
System.out.print("Enter no. of Term");
Scanner r = new Scanner(System.in);
n = r.nextInt(); // n=10

for(i=0; i<=n; i=i+2) // i=0 2 4 6 8 10
{
 if(i<=10)
 S.o.p(i); // 0 2 4 6 8 10
 i+=2
}

```
1      /* Print sum of Odd OR Even in Given Range */
2 // Even 10 (0+2+4+6+8+10=30)
3 // Odd 11 (1+3+5+7+9+11=36)
4
5 import java.util.Scanner;
6 class OddEven
7 {
8     public static void main(String[] args) {
9
10         int n,i,sum=0;
11         System.out.print("Enter Range ");
12         Scanner r=new Scanner(System.in);
13         n=r.nextInt();
14
15         if(n%2==0)
16         {
17             for(i=0;i<=n;i=i+2)
18             {
19                 sum=sum+i;
20             }
21             System.out.print("Sum Of Even Number "+sum);
22         }
23         else
24         {
25             for(i=1;i<=n;i=i+2)
26             {
27                 sum=sum+i;
28             }
29             System.out.print("Sum Of Odd Number "+sum);
30         }
31     }
32
33 }
34
35 }
```

```
charPrint.java
1      /* Print character in Java */
2 // input 10--->10
3 // input A---->A
4
5 import java.util.Scanner;
6 class charPrint
7 {
8     public static void main(String[] args) {
9
10     char ch;
11     System.out.print("Please Enter Character ");
12     Scanner r=new Scanner(System.in);
13     ch=r.next().charAt(0); //A
14     System.out.print(ch);
15
16 }
17 }
```

```
1      /* Print character in Java */
2 // input 10--->10
3 // input A---->A
4
5 import java.util.Scanner;
6 class charPrint
7 {
8     public static void main(String[] args) {
9
10     char ch;
11     System.out.print("Please Enter Character ");
12     Scanner r=new Scanner(System.in);
13     ch=r.next().charAt(2); //A
14     System.out.print(ch);
15
16 }
17 }
```

```
1         /* Vowel or Consonant in Java */
2 // input a e i o u ---> vowel
3 // input b c d etc...
4
5 import java.util.Scanner;
6 class Check
7 {
8     public static void main(String[] args) {
9
10     char ch;
11     System.out.print("Enter any Character ");
12     Scanner r =new Scanner(System.in);
13     ch=r.next().charAt(0);
14
15     if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
16     {
17         System.out.print("Vowel"); 
18     }
19     else
20     {
21         System.out.print("Consonant");
22     }
23 }
24 }
```

```
ASCI.java
1             /* Print ASCII value of character */
2 // input A ----> 65
3
4 import java.util.Scanner;
5 class ASCII
6 {
7     public static void main(String[] args) {
8
9         char ch;
10        System.out.print("Enter any Character ");
11        Scanner r = new Scanner(System.in);
12        ch=r.next().charAt(0); //A
13
14        int a = ch; //      A|
15        System.out.print("ASCII value of "+ch +" is "+ a);
16    }
17 }
```

```
ASCI.java
1             /* Print ASCII value of character */
2 // input A ----> 65
3
4 import java.util.Scanner;
5 class ASCII
6 {
7     public static void main(String[] args) {
8
9         char ch;
10        System.out.print("Enter any Character ");
11        Scanner r = new Scanner(System.in);
12        ch=r.next().charAt(0); //A
13
14        int a = ch; //    A|
15        System.out.print("ASCII value of "+ch +" is "+ a);
16    }
17 }
```

```
1             /* Maximum number b/w two Numbers */  
2 // input 10 20 ----> 20  
3 import java.util.Scanner;  
4 class Max  
5 {  
6     public static void main(String[] args) {  
7         int a,b;  
8         System.out.print("Enter two Number");  
9         Scanner r=new Scanner(System.in);  
10        a=r.nextInt();  
11        b=r.nextInt();  
12        if(a>b) // 10>20  
13        {  
14            System.out.print(a);  
15        }  
16        else  
17        {  
18            System.out.print(b);  
19        }  
20    }
```

```
Select Command Prompt  
Microsoft Windows [Version 10.0.18363]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\Users\WIN10>cd desktop  
C:\Users\WIN10\Desktop>  
C:\Users\WIN10\Desktop>java Max  
20  
C:\Users\WIN10\Desktop>
```