**Control Statements: Which path u should follow. (decision box)**

If-else

If

If- else if- else

Switch

**Loops: If u want to execute your block of code repeatedly**

1. for : simple, for-each
2. Do while
3. While

**For loop:**

for( initialization; condition; expression){

}

All the three initialization, condition and expression are optional in a for loop

**Table of 3:**

For(int i= 1; i<=10; i++){

Sysout( “3 \* “ + i + “ = “+ (3\*i));

}

Step 1: i= 1;

Step 2: i<=10;//if this condition is true, it will come inside the loop.

Step 3: Next thing would be to go the iteration step (i++)

Step 4: again check the condition. If condition is true, it will again come inside the loop

Step 5: goto step 3. If condition is false, come out of the loop.

**OR**

Int x= 1;

For( ;i<=10; i++){

}

**OR**

Int x=1;

for(; ; ){

If(x<=10){

X++;

}else{

break; //it takes it out of the loop

}

}

Q) for( ; ; ){

Sysout(“\*”);

}

Q) for(int i=0,j=0,k=1; i<=10 && j<=5 && k<=20; i++, j--){}

Q) \*

\* \*

\* \* \*

for(int i=1; i<=3; i++){

for(int j=1; j<=i;j++){

System.out.print(“\* “);

}

System.out.println();

}

**While loop**

While(condition){ //execute the loop till/while condition is true

//execution steps

}

Int i=0;

While(i<10){

Sysout(++i);

}

**private** **static** **void** printLoop() {

**int** i = 1;

**while** (i <= 3) {

**int** j = 1;

**while** (j <= i) {

System.***out***.print("\* ");

j++;

}

i++;

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

}

}

**Do While:**

This will first execute the code and then check the condition

do{

//statements

}while(condition);

int i=10;

do{

Sysout(i);

}while(i<10);

//Output: 10

Q) 3.6.9.12……30

**Vararg: Any number of integer arguments**

Void add(int… b){

Sysout(a+b);

}

Add(3,4), add

**public** **class** Demo1 {

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

*add*(3,4);

*mul*(3.4f);

}

**static** **void** mul(**float** b,**int**... a ) {

**int** sum=0;

**for**(**int** i=0; i<a.length;i++) {

sum= sum\* a[i];

}

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

}

**static** **void** add(**int** b, **int** c,**int**... a) {

**int** sum=0;

**for**(**int** i=0; i<a.length;i++) {

sum= sum+ a[i];

}

System.***out***.println(sum+b+c);

}

}

Vararg: 0 or more arguments

**For each loop**

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

**int**[] arr= {2,4,6,7,1,3};

String[] sarr= {"apple","orange","pear"};

**for**(**int** i=0; i<arr.length; i++) {

System.***out***.println(arr[i]);

}

**for**(**int** elem: arr) {

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

}

**for**(String elem: sarr) {

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

}

}

**import**

package p1;

class A{} interface C{}

package p2;

class A{}

package p3;

import p2.A;

import p1.C;

//If it is in a diff package, u shud import the class.

class B extends A implements C{}

**static import**

package p1;

class A{

static int i=5;

static void print(){};

}

Class B{

public static void main(String args[]){

A.print();

}

}

Import static A.print; //importing static fields(method/property) into ur class.

Import static A.i;

Class B{

public static void main(String args[]){

print();

Sysout(i);

}

}

**Final:** Once declared u cannot change it.

1. Variable: local, instance, class
2. Method
3. class

void m1(){

final int i= 5;

i++; //compile time error

}

Final instance variable: should be provided a value before an object is created.

Class A{

final int x; //error

}

Class A{

Final int x= 5;

}

Class A{

Final int x;

{

X= 5;

}

}

Class A{

Final int x;

A(){

X= 5;

}}

Class A{

Static final int x; //error

}

Class variables if final, should be initialized before the class is loaded

Class A{

Static final int x=5;

}

Class A{

Static final int x;

Static{

X=5;

}

}

**Final Method**

Final methods cannot be overridden

Class A{

Void m1(){}

}

Class B extends A{

Void m1(){} //error

}

**Final Class**

U cannot inherit from it or u cannot extend it or u cannot create a child class for it

Final Class A{}

Class B extends A {} //error

**String class**

Immutable class in java: Once u create the object of this class, u cannot change the state of the object.

final Class A{

private Int i;

A(int i){

this.i= I;

}

}

/\*Class B extends A{ int x;

B(int i){super(i);}

void print(){ x++;}

}\*/

B b= new B(5);

final A obj= new A(4); //i= 4

// changed the state of the object

* A is a mutable class.

Q) Make A as immutable class: Initialize once, but not change it thereafter.

final Class A{

private Int[] arr;

private Int x;

A(int x, int[] arr){ this.x= x; this.arr= arr;

}

Public int getX(){ return x;}

Public int[] getArr(){

Int[] arr1= new int[arr.length];

For(int I; i< arr.length; i++){

arr1[i]= arr[i];

}

return arr1;}

}

Int[] arr= {2,3,4,5};

A obj= new A(5, arr);

Int[] arr1=Obj.getArr();

Arr1[2]= 45;

Sysout(arr[2]); //45