Looping statements:

1. For loop:

For(statement; cond; statement){}

Infinite loop: for( ; ; )

Or for(; true;)

Eg. Of continue;

**package** com.te.TYbatch.training;

**import** java.util.Scanner;

**public** **class** Demo {

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

**for**(**int** i = 1; i< 10 ; i++) {

**if**(i % 2 == 0) {

**continue**;

}

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

}

}

}

**package** com.te.TYbatch.training;

**public** **class** Demo {

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

**for**(**int** i=1; i<4;i++) {

System.***out***.println("hello"+ i);

}

}

}

**package** com.te.TYbatch.training;

**public** **class** Demo {

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

**for**(**int** i= 1; i<=20 ; i++) {

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

i+=i;

}

}

}

**package** com.te.TYbatch.training;

**public** **class** Demo {

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

**int** i = 0;

**for**(System.***out***.println("hello"); i<5 ; System.***out***.println("hi")) {

i++;

}

}

}

1. While(cond){

}

1. Do{

Statement //will always execute atleast once

While(cond){}

1. For each loop()🡪 it has to be an array or an iterable type

Making Static reference in a non static/static method is possible. But making a non-static ref in a static method not possible

**package** com.te.TYbatch.training;

**public** **class** Demo {

**static** **int** *j*;

**int** i;

**public** **static** **void** main() {

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

System.***out***.println(i); //error

}

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

Demo demo = **new** Demo();

demo.*main*();

}

}

METHODS AN ARGUMENTS.

**package** com.te.TYbatch.training;

**public** **class** Demo {

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

*m1*(1,2); //actual args

}

**static** **void** m1(**int** a, int b) { // formal args

System.***out***.println("M1 is executing");

}

}

Var args(int…a)