**This and Super**

**1. Print the fields/instance members of the current class using this and without using object**

class Test

{

int a;

int b;

// Parameterized constructor

Test(int a, int b)

{

this.a = a;

this.b = b;

}

void display()

{

//Displaying value of variables a and b

System.out.println("a = " + a + " b = " + b);

}

public static void main(String[] args)

{

Test object = new Test(10, 20);

object.display();

}

}

**2. Print the fields/instance members of the parent class using super**

class Animal{

String color="white";

}

class Dog extends Animal{

String color="black";

void printColor(){

System.out.println(color);//prints color of Dog class

System.out.println(super.color);//prints color of Animal class

}

}

class TestSuper1{

public static void main(String args[]){

Dog d=new Dog();

d.printColor();

}}

**3. Call constructor of the current class using this()**

class Student{

int rollno;

String name;

float fee;

Student(int rollno,String name,float fee){

rollno=rollno;

name=name;

fee=fee;

}

void display(){System.out.println(rollno+" "+name+" "+fee);}

}

class TestThis1{

public static void main(String args[]){

Student s1=new Student(111,"ankit",5000f);

Student s2=new Student(112,"sumit",6000f);

s1.display();

s2.display();

}}

**4. Call argument constructor of current class using this()**

**5. Call constructor of the parent class using super()**

class Animal {

// default or no-arg constructor of class Animal

Animal() {

System.out.println("I am an animal");

}

}

class Dog extends Animal {

// default or no-arg constructor of class Dog

Dog() {

// calling default constructor of the superclass

super();

System.out.println("I am a dog");

}

}

class Main {

public static void main(String[] args) {

Dog dog1 = new Dog();

}

}

**6. Use this() and super() in methods not in constructors**

class Animal {

String name;

Animal(String name) {

this.name = name;

}

public void move() {

System.out.println("Animals can move");

}

public void show() {

System.out.println(name);

}

}

class Dog extends Animal {

Dog() {

//Using this to call current class constructor

this("Test");

}

Dog(String name) {

//Using super to invoke parent constructor

super(name);

}

public void move() {

// invokes the super class method

super.move();

System.out.println("Dogs can walk and run");

}

}

public class Tester {

public static void main(String args[]) {

// Animal reference but Dog object

Animal b = new Dog("Tiger");

b.show();

// runs the method in Dog class

b.move();

}

}