Polymorphism:

Polymorphism means one name many forms. Any things that exists in more than one form is known as a polymorphism. Polymorphism applies only to non-private instance method it does not apply to variable or static method. Inherited method in the case of polymorphism cannot private. There are two types of polymorphism one is compile time polymorphism and the other is run time polymorphism.

Compile time polymorphism is function and operators overloading. Runtime polymorphism is done using Inheritance

Overriding and Overloading:

Method overriding required the same method signature and the same return type. Only non-final instance methods in the superclass that are directly accessible from the subclass are eligible for overriding.

Overloading occurs when the method names are the same, but the parameters lists differ. Therefore to overloads methods, the parameters must differ in type, order or number. As the return type is not a part of the signature , having different return types is not enough to overload methods.

* Method overloading is possible in same class and in case of inheritance.
* Method overriding is possible in only case of inheritance .
* When sub and super class both have method with the same name, same argument and same return type it is called method overriding.
* If return type is not same in the case of method overriding compile time error will be occur.
* In the case of method overriding, the overridden method of sub class can be less restricted, but it cannot be more restricted.
* Ordinary method can inherit but the constructer of a class never inherited regardless their attribute.
* Static method cannot overridden.

class One {

private int x;

public void set( int a){

x=a;

}

public void show() {

System.out.println(“X=”+x);

}

}

class Two extends One {

int x;

public void set(int a, int b){

{

set(a);

x=a;

}

public void show(){

super.show();

System.out.println(“Y=”+x);

}

}

class Use{

public static void main(String[] args){

obj.set(10, 20);

obj.show();

}

}

Output :

X = 10

Y = 10;