1. Single inheritance

class Parent {

void show() {

System.out.println("Parent class method");

}

}

class Child extends Parent {

void display() {

System.out.println("Child class method");

}

}

public class SingleInheritance {

public static void main(String[] args) {

Child obj = new Child();

obj.show();

obj.display();

}

}

Output :

Parent class method

Child class method

1. Multiple inheritance

Interface InterfaceA {

Void methodA();

}

Interface InterfaceB {

Void methodB();

}

Class MultipleInheritanceDemo implements InterfaceA, InterfaceB {

Public void methodA() {

System.out.println(“Method from InterfaceA”);

}

Public void methodB() {

System.out.println(“Method from InterfaceB”);

}

}

Public class MultipleInheritance {

Public static void main(String[] args) {

MultipleInheritanceDemo obj = new MultipleInheritanceDemo();

Obj.methodA();

Obj.methodB();

}

}

Output:

Method from InterfaceA

Method from InterfaceB

1. Multilevel inheritance

Class Grandparent {

Void grandparentMethod() {

System.out.println(“Grandparent class method”);

}

}

Class Parent extends Grandparent {

Void parentMethod() {

System.out.println(“Parent class method”);

}

}

Class Child extends Parent {

Void childMethod() {

System.out.println(“Child class method”);

}

}

Public class MultilevelInheritance {

Public static void main(String[] args) {

Child obj = new Child();

Obj.grandparentMethod();

Obj.parentMethod();

Obj.childMethod();

}

}

Output:

Grandparent class method

Parent class method

Child class method

1. Hybrid inheritance

Interface InterfaceA {

Void methodA();

}

Interface InterfaceB {

Void methodB();

}

Class Parent {

Void parentMethod() {

System.out.println(“Parent class method”);

}

}

Class HybridInheritanceDemo extends Parent implements InterfaceA, InterfaceB {

Public void methodA() {

System.out.println(“Method from InterfaceA”);

}

Public void methodB() {

System.out.println(“Method from InterfaceB”);

}

}

Public class HybridInheritance {

Public static void main(String[] args) {

HybridInheritanceDemo obj = new HybridInheritanceDemo();

Obj.parentMethod();

Obj.methodA();

Obj.methodB();

}

}

Output:

Parent class method

Method from InterfaceA

Method from InterfaceB

1. Hierarchical inheritance

Class Parent {

Void parentMethod() {

System.out.println(“Parent class method”);

}

}

Class Child1 extends Parent {

Void child1Method() {

System.out.println(“Child1 class method”);

}

}

Class Child2 extends Parent {

Void child2Method() {

System.out.println(“Child2 class method”);

}

}

Public class HierarchicalInheritance {

Public static void main(String[] args) {

Child1 obj1 = new Child1();

Obj1.parentMethod();

Obj1.child1Method();

Child2 obj2 = new Child2();

Obj2.parentMethod();

Obj2.child2Method();

}

}

Output:

Parent class method

Child1 class method

Parent class method

Child2 class method

Method overriding

Class Parent {

Void show() {

System.out.println(“This is the parent class method”);

}

}

Class Child extends Parent {

@Override

Void show() {

System.out.println(“This is the overridden method in the child class”);

}

}

Public class MethodOverridingExample {

Public static void main(String[] args) {

Parent obj1 = new Parent();

Obj1.show();

Parent obj2 = new Child();

Obj2.show();

}

}

Output:

This is the parent class method

This is the overridden method in the child class