Interface in Java

```
Interface one{

Public void eat(); //abstract method no

Body

Interface two{

Public void bark(); //abstract method no

Body

}

Interface three extends one,two{

Public void meow(); //abstract method no body

}

Class ChildClass implements one, two{

Public void eat() { System.out.println("ChildClass is eating"); }

Public void bark() { System.out.println("ChildClass is barking"); }

Public static void main(String[] args) {

// Create an object o
Public void bark(); //abstract method no body }
Interface three extends one,two{ public void meow(); //abstract method no
body }

Class ChildClass implements one,two{ public void eat() {
System.out.println("ChildClass is eating"); }

Public void bark() {

System.out.println("ChildClass is barking"); }

Public static void main(String[] args) {

// Create an object o

ChildClass c = new ChildClass();

c.eat();

c.bark();

}
  }
}
```

Multiple Interface:

```
Interface A {

Void showA();

}
Interface B {
```

```
Void showB();

}

Class C implements A, B {

Public void showA() {

System.out.println("This is from Interface

A.");

}

Public void showB() {

System.out.println("This is from Interface

B.");

}

}

Public class Main {

Public static void main(String[] args) {

C obj = new C();

Obj.showA();
Obj.showB();
    }
}
```

Hierarchical:

```
Class Animal {

Void eat() { System.out.println("Animal eats food."); }

}

Class Dog extends Animal {

Void bark() { System.out.println("Dog barks."); }

}

Class Cat extends Animal {

Void meow() { System.out.println("Cat meows."); }

}

Public class Main {

Public static void main(String[] args) {

Dog d = new Dog();

d.eat();
d.bark();

Cat c = new Cat();

c.eat();

c.meow();

}

}
```

```
/ Interface 1

Interface Engine {

Void engine Type();

}

// Interface 2

Interface Electric {

Void batteryCapacity(); }

// Parent class

Class Vehicle {

Void wheels() {
```

System.out.println(“This vehicle has 4 wheels.”); } }

// Child class with hybrid inheritance

Class ElectricCar extends Vehicle implements

Engine, Electric {

Public void engineType() {

System.out.println(“Engine: Electric Motor”); }

Public void batteryCapacity() {

System.out.prin

----

hierarchical inheritance example

// Parent Class (Superclass)

Class Animal {

Void eat() {

System.out.println(“Animal is eating”); }

// Child Class 1

Class Dog extends Animal {

Void bark() {

```
System.out.println(“Dog is barking”); }

}

// Child Class 2

Class Cat extends Animal {

Void meow() {

System.out.println(“Cat is meowing”); }

}

// Main Class to Run the Code

Public class Hierarchical Example {

Public static void main(String[] args) {
```
