

Abstraction

```
interface Engine {  
    void startEngine ();  
    void stopEngine (); }  
interface Driveable {  
    void drive (); }  
abstract class Vehicle {  
    String brand;  
    public Vehicle (String brand) {  
        this.brand = brand;  
    }  
    abstract void fuelType ();  
    public void showBrand () {  
        System.out.println ("Vehicle Brand : " + brand);  
    }  
}
```

```
class Car extends Vehicle implement Driveable, Engine  
{  
    public Car (String brand) {  
        super (brand); }  
}
```


@Override

```
public void drive () {
```

```
    System.out.println ("Car is driving on the road.");
}
```

@Override

```
void fuelType () {
```

```
    System.out.println ("Car uses petrol.");
}
```

@Override

```
public void startEngine () {
```

```
    System.out.println ("Car engine started.");
}
```

@Override

```
public void stopEngine () {
```

```
    System.out.println ("Car engine stopped.");
}
```

```
class ElectricBike extends Vehicle implements Driveable, Engine
```

```
{
    public ElectricBike (String brand) {
```

```
        super (brand);
    }
```


@Override

```
public void drive () {  
    System.out.println ("Electric bike is gliding  
        silently.");  
}
```

@Override

```
void fuelType () {  
    System.out.println ("Electric bike uses battery  
        power.");  
}
```

@Override

```
public void startEngine () {  
    System.out.println ("Electric bike powered on.");  
}
```

@Override

```
public void stopEngine () {  
    System.out.println ("Electric bike powered off.");  
}
```


class Boat extends Vehicle implements Driveable, Engine {

public Boat (String brand) {

super (brand); }

@Override

public void drive () {

System.out.println ("Boat is sculling through water.");

}

@Override

void fuelType () {

System.out.println ("Boat uses diesel fuel."); }

@Override

public void startEngine () {

System.out.println ("Boat engine roars to life."); }

@Override

public void stopEngine () {

System.out.println ("Boat engine shut down."); }


```

public class Abstraction {
    public class (brand, price) {
    public static void main (String[] args) {

```

```

    Vehicle car = new Car ("Toyota");

```

```

    car.showBrand();

```

```

    ((Engine) car).startEngine();

```

```

    ((Driveable) car).drive();

```

```

    car.fuelType();

```

```

    ((Engine) car).stopEngine();

```

```

    System.out.println();

```

```

    Vehicle ebike = new ElectricBike ("Tesla");

```

```

    ebike.showBrand();

```

```

    ((Engine) ebike).startEngine();

```

```

    ((Driveable) ebike).drive();

```

```

    ebike.fuelType();

```



```
((Engine) ebike). stopEngine ();
```

```
system.out.println ();
```

```
Vehicle boat = new Boat ("Yamaha");
```

```
boat.showBrand ();
```

```
((Engine) boat). startEngine ();
```

```
((Driveable) boat). drive ();
```

```
boat.fuelType ();
```

```
((Engine) boat)
```

```
((Engine) boat). stopEngine ();
```

```
}  
}
```