

```
interface Drive{
```

```
    void drive();
```

```
}
```

```
interface Engine{
```

```
    void startEngine();
```

```
    void stopEngine();
```

```
}
```

```
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 and extends vehicle implements Drive, Engine{
```

```
    public car(String brand){
```

```
        super(brand);
```

```
    }
```

@Override

```
public void drive() {  
    System.out.println("car is driving on the road.");  
}
```

@Override

```
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 Drive, Engine {
    public class ElectricBike (String brand) {
        super (brand);
    }
    @Override
    public void drive () {
        System.out.println ("Electric Bike is gliding silently.");
    }
    public void fuelType () {
        System.out.println ("Electric Bike uses battery power.");
    }
    public void startEngine () {
        System.out.println ("Electric Bike powered on.");
    }
    public void stopEngine () {
        System.out.println ("Electric Bike powered off.");
    }
}

```

```
class Boat extends Vehicle implements Drive, Engine {
```

```
    public Boat(String brand) {
```

```
        super(brand);
```

```
    }
```

```
    @Override
```

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

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

```
    }
```

```
    @Override
```

```
    public void fuelType() {
```

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

```
    }
```

```
    @Override
```

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

```
        System.out.println("Boat engines roars to life.");
```

```
    }
```

```
    @Override
```

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

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

```
    }
```

```
}
```

```
public class Abstract {  
    public static void main (String [] args) {  
        Vehicle car = new Car ("Toyota");  
        car.showBrand ();  
        ((Engine)car).startEngine ();  
        ((Drive)car).drive ();  
        car.fuelType ();  
        ((Engine)car).stopEngine ();  
        System.out.println ();  
        Vehicle ebike = new ElectricBike ("Tesla");  
        ebike.showBrand ();  
        ((Engine)ebike).startEngine ();  
        ((Drive)ebike).drive ();  
        ebike.fuelType ();  
        ((Engine)ebike).stopEngine ();  
        System.out.println ();  
        Vehicle boat = new Boat ("Yamaha");  
        boat.showBrand ();
```

```
        ((Engine) boat).startEngine();  
        ((Drive) boat).drive();  
        boat.fuelType();  
        ((Engine) boat).stopEngine();  
    }  
}
```