

IT-22010

### Abstract Practice

```
interface Engine {
```

```
    void startEngine();    // no body
```

```
    void stopEngine();
```

```
}
```

```
abstract class Vehicle {
```

```
    String brand;
```

```
    int year;
```

```
    public Vehicle (String brand, int year) {
```

```
        this.brand = brand;
```

```
        this.year = year;
```

```
}
```

```
    abstract void drive();
```

```
    public void displayInfo() {
```

```
        System.out.println("Brand: " + brand + " Year: " + year);
```

```
}
```

```
}
```

class Car extends Vehicle implements Engine {

public Car (String brand, int year) {

super (brand, year);

}

public void startEngine () {

System.out.println ("Car engine start");

}

public void stopEngine () {

System.out.println ("Car engine stop");

}

void drive () {

System.out.println ("Driving a car");

}

}

class Bike extends Vehicle implements Engine {

public Bike (String brand, int year) {

super (brand, year);

}

public void startEngine () {

System.out.println ("Bike engine start");

}

```
public void stopEngine() {  
    system.out.println("Bike Engine stop");  
}
```

```
void drive() {  
    system.out.println("Riding a bike");  
}
```

```
public class AbstractPractice {  
    public static void main (String[] a) {  
        Vehicle car = new Car ("Mercedes", 2022);  
        car.displayInfo();  
        ((Engine).car).startEngine();  
        car.drive();  
        ((Engine).car).stopEngine();  
        system.out.println();  
        Vehicle bike = new Bike ("Suzuki", 2023);  
        bike.displayInfo();  
        ((Engine).bike).startEngine();  
        bike.drive();  
        ((Engine).bike).stopEngine();  
    }  
}
```