

ID: IT: 22044 Anik Mollik

import java.io.PrintStream

// Interface: defines engine vehicle's behaviour
that all vehicles must implement

interface Engine {

PrintStream out = System.out; // Allowed: constant field

void startEngine();

void stopEngine();

} // interface most different code: no fields

// Abstract class: base class for all vehicles.

Abstract class Vehicle {

String brand;

int year;

{ constructor: sets brand and year

public Vehicle (String brand, int year) {

this.brand = brand;

this.year = year;

}

// Abstract method: must be implemented by all
 // sub classes

abstract void drive();

// concrete method: shared behavior for
 // all vehicles

public void displayInfo() {

 Engine.o.println ("Brand: " + brand + " Year:
 // " + year);

// subclass: Car inherits from vehicles and
 // implements Engine interface

class Car extends Vehicle implements Engine {

// constructor: calls superclass constructor to set
 // brand and year

public Car (String brand, int year) {
 super (brand, year);

}

// implement start Engine method
 // from Engine interface.

```
@Override  
public void startEngine() {  
    System.out.println("Car Engine started");  
}
```

//implement abstract drive method from vehicle class

```
@Override  
void drive() {  
    System.out.println("Driving a car");  
}
```

//sub class Bike inherits from vehicle and implements Engine interface

```
class Bike extends Vehicle implements Engine {  
    public Bike(String brand, int year) {  
        super(brand, year);  
    }  
}
```

//implement startEngine method from Engine interface

```
@Override  
public void startEngine() {  
    System.out.println("Bike Engine started");  
}
```

```
@Override  
public void stopEngine() {  
    o.println("Bike Engine stopped"); }  
// implement abstract drive method from  
// vehicle class  
@Override  
void drive() {  
    o.println("Riding a bike"); }  
// main class  
public class TestVehicle {  
    public static void main(String[] args) {  
        Engine o.println("Name : Aria MFD : IT 22084");  
        vehicles can new Car("Toyota" 2022);  
        can displayInfo();  
        ((Engine) can).startEngine(); } }
```

car. drive();

((Engine) car). stopEngine();

Engine. o.println();

vehicle bike = new Bike ("Yamaha", 202

bike. displayInfo();

((Engine) bike). startEngine();

bike. drive()

((Engine) bike). stopEngine();

}