

```
import java.io.*;
interface EngineControl {
    PrintStream o = System.out;
    void startEngine();
    void stopEngine();
}
```

```
abstract class Transport {
    String make;
    int manufactureYear;
    public Transport(String make, int manufactureYear) {
        this.make = make;
        this.manufactureYear = manufactureYear;
    }
    abstract void operate();
```

```
    public void showDetails() {
        EngineControl o.println("Make: " + make +
                                " + manufactureYear: " + manufactureYear);
    }
}
```

```
class Sedan extends Transport implements EngineControl {
    // ...
}
```

```
public Sedan(String make, int manufactureYear){
    super(make, manufactureYear);
}
```

Exemple de

public void stopEngineOf  
start

```
o.println("Sedan engine started.");
```

② override

```
@Override
public void stopEngine() {
    // Stop the engine
}
```

```
public void stopEngine() {
    o.println("Sedan engine stopped.");
}
```

@Override

```
void operate() {
```

```
void operate() {
    o.print("\nDriving the Sedan.");
}
```

2

3

```
}  
}  
@given class Motorbike extends Transport implements  
    EngineControl {  
    like (String make int max)
```

```

}
over class Motorbike extends Machine {
    public Motorbike(String make, int manufactureYear) {
        EngineControl {

```

Super (make, manufacture ~~date~~ year) { ; }



TOPIC NAME: \_\_\_\_\_

DAY: \_\_\_\_\_

TIME: \_\_\_\_\_

DATE: \_\_\_\_\_

@Override

```
public void startEngine() {  
    o.println("Motorbike engine started.");  
}
```

@Override

```
public void stopEngine() {  
    o.println("Motorbike engine stopped.");  
}
```

@Override

```
void operate() {  
    o.println("Riding the motorbike.");  
}
```

}

```
}  
public class TransportDemo {  
    public static void main(String[] args) {  
        EngineControl o = new EngineControl("Name: Sayam\nID: 112208\n");  
        Transport sedan = new Sedan("Honda", 2024);  
        sedan.showDetails();  
        ((EngineControl) sedan).startEngine();  
        sedan.operate();  
        ((EngineControl) sedan).stopEngine();  
    }  
}
```

EngineControl.openFuel();

newport motorcycle = new Motorcycle();  
motorbike = startPedals();

(EngineControl) motorcycle, start Eng

motorbike.openate();

(EngineControl) motorcycle). start Eng

}

}