A Flashlight's method `makeLight()` returns true or false depending on whether its BatteryState is Good or Dead.

```java
public class Flashlight() {
    private Good good;
    private Dead dead;
    private State state;

    public Flashlight() {
        good = new Good();
        dead = new Dead();
        state = good;
    }

    public boolean makeLight() { return state.makeLight(); }  

    public void setDead() { state = dead; }
}

abstract public class State {
    abstract public boolean makeLight();
}

public class Good extends State {
    public boolean makeLight() { return true; }
}

public class Dead extends State {
    public boolean makeLight() { return false; }
}
```

1) [70%] Finish writing Flashlight using the **State Pattern**. Do not use any conditionals (e.g. 'if') or 'instanceof'.

2) [30%] On back, draw the **UML class diagram**, with all classes, methods, label which are abstract, and all "has-a" instances.