

# Honor Pledge

On my honor, I have neither given nor received any unauthorized aid on this quiz.

By typing your first and last name in the space provided below you are electronically signing to indicate that:

- (1) You are the person who is taking this quiz.
- (2) You read and understood the Honor Pledge and you agree to be bound by it.

**Question 1.** (4 pts) Consider the following code which implements the Observer pattern:

```
import java.util.*;
interface Observable {
    public void notifyObservers();
    public void addObserver(Observer o);
}
interface Observer {
    public void update(int i);
}
class SoundEffectController implements Observer{
    private boolean isSoundPlaying = false;
    public void playSound(){
        System.out.println("Play a sound");
        isSoundPlaying = true;
    }
    public void update(int yCoordinate) {
        if(yCoordinate > 0 && isSoundPlaying == false){
            playSound();
        }
    }
}
class LightEffectController implements Observer{
    private boolean isLightFlickering = false;
    public void flickerLights(){
        System.out.println("Flicker the lights");
        isLightFlickering = true;
    }
    public void update(int xCoordinate) {
        if(xCoordinate > 10 && isLightFlickering == false){
            flickerLights();
        }
    }
}
class Player implements Observable{
    private int xCooridinate;
    private int yCooridinate;
    private List<Observer> observers = new ArrayList<Observer>();
    public void move(int steps){
        xCooridinate += steps;
        notifyObservers();
    }
    public void jump(int height){
        yCooridinate += height;
        notifyObservers();
    }
    public void notifyObservers() {
        for(Observer o:observers){
            if(o instanceof SoundEffectController){
                ((SoundEffectController)o).update(yCooridinate);
            }
            else{
                if(o instanceof LightEffectController){
                    ((LightEffectController)o).update(xCooridinate);
                }
            }
        }
    }
    public void addObserver(Observer o) {
        this.observers.add(o);
    }
}
```

Specify which model this example follows:

Question 2. (3 pts) Refactoring:

**Question 3.** (3 pts) Which refactoring eliminates uninformative variables?