Model 1 Abstract Methods

The `abstract` keyword can be used to declare methods that have no body. Classes with abstract methods must also be defined as abstract.

```
public abstract class LoudToy {
    private int volume;

    public LoudToy(int volume) {
        this.volume = volume;
    }

    public int getVolume() {
        return volume;
    }

    public void setVolume(int volume) {
        this.volume = volume;
        makeNoise();
    }

    public abstract void makeNoise();
}
```

Questions (15 min) Start time:

1. Summarize the differences between Model 1 and your answer to **??**.

2. Open *LoudToy.java* (from Model 1) in your IDE. Remove the word `abstract` from the class definition. What are the two compiler errors?

3. Replace the word `abstract` in the class definition, and then remove the word `abstract` from the method definition. What is the compiler error now?

4. Remove the definition of makeNoise altogether, and notice the compiler error. Why is it necessary to declare this method in LoudToy?

5. Undo all changes in *LoudToy.java*, and add the following main method. What is the compiler error message? Why do you think Java doesn't allow you to construct a LoudToy?

```
public static void main(String[] args) {
    LoudToy toy1 = new LoudToy(1);
    toy1.makeNoise();
}
```

6. Open *ToySheep.java* and rename makeNoise to makeNoise2. What is the compiler error?

7. Rename the method back to makeNoise, but change void to int. What is the error now?

8. Explain how an abstract method is like a contract.