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 **??**.

The class and the `makeNoise` method are declared as abstract. The definition of `makeNoise` ends with a semicolon, rather than an empty body `{}`.

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

The type LoudToy must be an abstract class to define abstract methods.

The abstract method makeNoise in type LoudToy can only be defined by an abstract class.

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?

This method requires a body instead of a semicolon.

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

The `setVolume` method calls the `makeNoise` method.

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();
}
```

The compiler says, "Cannot instantiate the type LoudToy." Abstract classes cannot be instantiated, because some of their methods aren't implemented.

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

The type ToySheep must implement the inherited abstract method LoudToy.makeNoise().

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

The return type is incompatible with LoudToy.makeNoise().

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

If you inherit an abstract class, you must override the abstract methods exactly as defined. This is important because they might be called in the code of the abstract class.