In-class exercises (lecture 2)

At the start of every lecture:

1. Pull the latest code from the lecture-code repo
2. Open the Evening_lectures folder
3. Copy this week's folder somewhere else so you can edit the code without causing GitHub conflicts
4. Open the code:
    1. Find the build.gradle file in the folder called LectureX
    2. Double click build.gradle to open the project in IntelliJ.

Exercise 1: Read some code

- Read through all the classes in the `bookexample` package. How are they related / used?
- Skim the test files.
- What details do you notice that we haven't talked about yet?

Exercise 2: Exceptions and testing

Your work on in-class does not need to be turned in. Don't worry too much about getting the solutions exactly right—the goal is to get some experience with new concepts. We will go over sample code right after the exercise and the completed code will be posted to the lecture-code repo after class.

In `bookexample > Account.java`:

- Add a private helper method that validates a password. The only requirement for a valid password is that it is at least 8 characters long.
    - To get the length of a String in Java: `variableName.length()`
- Add a public method, `void resetPassword(String password)`, that allows the password value to be changed.
- Use the helper method to throw an `IllegalArgumentException` if a password is invalid.
    - Syntax: `throw new IllegalArgumentException("message here");`
    - *Note: IllegalArgumentException is an unchecked exception and therefore does not need you to add "throws IllegalArgumentException" to the method signature. In general, you will be writing your own custom checked exceptions, which will require this addition. You'll see how to write your own exceptions later today.*
- In AccountTest.java, add unit tests to confirm that passwords cannot be created / reset if the length is invalid.
    - To test that an exception is actually thrown when it's supposed to be thrown, add the

following code to the @Test notation above the appropriate test method: `(expected=ExceptionClassName.class)`. Replace "ExceptionClassName" with the type of exception that should be thrown.

Exercise 3: Inheritance

In the `inheritanceexample` package:

- Write a **test class** for the `Student` class
  - Use Gradle to view coverage of inherited methods. Tip: If you find yourself wondering if you need to test a method on an assignment, the coverage report may have the answer.
- Write an `Instructor` class that also inherits the `Person` class
  - You can decide what additional properties / methods the class should have.
- Generate a **UML diagram** that shows all three classes (`Person`, `Student`, `Instructor`).
  - **IntelliJ docs on UML diagrams (https://www.jetbrains.com/help/idea/class-diagram.html)**

Extension: exception practice

In the `bookexample` package > `Book` class, throw a custom exception if the provided ISBN is invalid.

- You will need to create the custom exception.
- A helper method for validating an ISBN is already in the `Book` class.
- Adding tests is also recommended.