

CS 145: Preassignment 2

On Security

1 Description

The goal of this preassignment is to give you some practice with conditionals and loops. You will do so in the context of developing some methods that test the strength of passwords. We want our passwords to be complex enough that those with malicious intent have a much harder time trying to crack it.

2 Requirements

To receive full credit for this assignment, you must satisfy the following requirements:

- Write all code in a class named `Password` in a package named `pre2`. Case and spelling matter.
- Compose a handful of `public` and `static` methods, described below, all of which take a `String` password as their sole argument. The password may be blank. Check out the methods in class `Character` to help you verify the criteria.
 - A method `isTrivial` that returns true if the password is “password” or the password is blank.
 - A method `containsUpper` that returns true if the password contains at least one uppercase letter and false otherwise.
 - A method `containsDigit` that returns true if the password contains at least one digit and false otherwise.
 - A method `containsNonAlphanumeric` that returns true if the password contains at least one character that is neither a letter nor a digit.
 - A method `containsWhitespace` that returns true if there is at least one whitespace character (tab, space, linebreak, and so on) in the password and false otherwise.
 - A method `containsAt` that returns true if an `@` is present in the password.
 - A method `isLegal` that verifies that the password is strong and in the proper format. It returns true if the password meets the following criteria: is non-trivial, contains an uppercase letter, contains a digit, contains a non-alphanumeric character, and contains neither whitespace nor an `@`. If any criteria is unmet, false is returned.

You may share testing code on the discussion board, but please do not share code from your `Password` class.

3 Submission

This is a preassignment, and preassignments are graded by SpecCheckers, not your instructor or TAs. The good news is you get to tweak your code until you have a perfect score. The bad news is that there's a little setup involved—and you must leave yourself enough time to work out kinks. Deadlines are not extended for botched configurations. Please follow these directions closely to test and submit your work:

1. Add JUnit to the Build Path if you haven't already done so. See preassignment 1 for details.
2. Download the SpecChecker JAR file.
3. Drag the JAR file onto your `pre2` package in Eclipse. Right-click on it, and select Build Path → Add to Build Path. It will migrate to the Referenced Libraries node in your project.
4. Run the JAR file by selecting it and hitting the Play button in the Eclipse toolbar. Run it as a Java application if prompted.
5. Address any errors. **The score you see will be the score you are assigned in the gradebook.**
6. Once you are satisfied with your score, choose a directory in which `pre2.zip` will be saved. This file contains a compressed version of `Password.java`.
7. Drop your submission into `W:\c s\CJohnson\cs145\<YOUR-USERNAME>`. You may overwrite this file as often as you like before the deadline. Just make sure the directory contains only one *.zip file with lowercase `pre2` in its name. Double-click on the file to make sure it's readable and uncorrupted.