
600.107 Introductory Programming in Java, Summer 2015
Homework 7
Due: Tuesday, July 21st at 11:59pm

Please review the **Homework Guidelines** handout for this course before you begin.

Restriction: For all tasks in this assignment, you are NOT permitted to use arrays or ArrayLists.

Repetition Statements: "while", "for", or "do-while" - type statements MAY be used in this exercise, but you are NOT permitted to use `break`; or `continue`; statements except that you may use `break`; inside a case within a `switch` statement.

Tasks 1 and 2. For this assignment, you'll be implementing an instantiable class called `GradeDist` and a driver program called `GradeDistDriver` which allows a user to keep track of the distribution of letter grades in a course. For this assignment, you are given javadoc-generated documentation for each class definition, and you need to implement each described data member, constructor, and method, exactly as specified in the documentation.

The PDF documentation for the `GradeDist` and `GradeDistDriver` classes is available on Piazza within a zip file named `HW7Files.zip`. Please read the descriptions of each member (i.e. data field, constructor, or method) carefully, as there are several implementation-related requirements for the assignment listed there. Two sample executions of the `main` method in the driver, and the data files used to create them are also available within `HW7Files.zip`.

Tips for Completing Tasks 1 and 2.

- (1) Implement the methods *exactly* as specified. For this assignment, it is not enough to write a completed project which can mimic the sample runs; you must actually write the methods described in the documentation, matching the functionality specified in the descriptions. You may add additional data members or private helper methods to either file if you wish, but you must implement all data members, constructors, and methods mentioned in the documentation.
- (2) In writing your classes, access modifiers such as `private` and `public` are important. In the documentation, methods which are private are listed with the modifier, but for methods which are public, no access modifier is shown. (That is, javadoc's notation is to imply that everything is public unless it says otherwise.) Add the modifier `public` in your code wherever it applies.
- (3) In the documentation, when a class like `String` is referenced, javadoc refers to it using its fully-qualified name, which is `java.util.String`. You don't need to write this in your code, just use `String` as the name of the class instead, as we always have done.
- (4) Practice code re-use where possible. If you have to write code to undertake essentially the same task more than two or three times, it generally makes sense to write one method that performs the task, and then call it repeatedly. In specifying methods for you on this assignment, we attempted to steer you to write helpful methods to enable good code reuse. So don't reinvent the wheel here; if you find yourself thinking, "I know how to write this; I already did it once in this project", then there's a very good chance you should be calling an existing method!

Turn-in. Before the due date listed at the top of this file, please submit via the course Blackboard site one complete `.zip` file named `HW7-jhed.zip` (where you replace 'jhed' with your own personal JHED) containing all `.java` files required for the above tasks. Be sure that all submitted code compiles! If you were unable to complete a task, please include as part of your `.zip` submission a text file named `README` explaining anything you'd like the graders to know.