P10 Course Registration

Overview

In this program, you will create a priority queue to help you organize the classes you wish to take in Spring 2023. When it's time to register, you'll be able to ask the queue for the top N credits worth of courses, ranked by whether they have seats available, whether they are required for your major, and the rating of the professor.

To make this as easy as possible, we'll be using an array-based max-heap to implement a priority queue.

Grading Rubric

5 points	**Pre-assignment Quiz**: accessible through Canvas until 11:59PM on 12/09.
15 points	**Immediate Automated Tests**: accessible by submission to Gradescope. You will receive feedback from these tests *before* the submission deadline and may make changes to your code in order to pass these tests. ⭐ All immediate tests for P10 will use ONLY your **TESTER** METHODS.⭐ Passing all immediate automated tests does **not** guarantee full credit for the assignment.
30 points	**Additional Automated Tests**: these will also run on submission to Gradescope, but you will not receive feedback from these tests until after the submission deadline.

Learning Objectives

After completing this assignment, you should be able to:

- **Create** a comprehensive testing suite for your own programs based on provided documentation
- **Explain** the functions of a priority queue and **implement** them using a heap
- **Incorporate** the Comparable interface into the program to facilitate PriorityQueueADT
- **Finish** CS 300 with no more programming assignments :)

Additional Assignment Requirements and Notes

Keep in mind:

- Pair programming is **NOT ALLOWED** for this assignment. You must complete the assignment individually.

- The ONLY external libraries you may use in your program are the Iterator interface and any exceptions.

- You are allowed to define any **local** variables you may need to implement the methods in this specification. You may NOT define any additional data fields or static fields.

- ALL test methods must be **public static**, take **zero** arguments, and return a **boolean** value. These methods MUST be contained within the CourseRegTester class.

- All classes and public methods must have their own Javadoc-style method header comments in accordance with the CS 300 Course Style Guide, and any methods longer than three lines should have *at least one* comment in the body of the method.

- Any source code provided in this specification may be included verbatim in your program without attribution.

- **Run your program locally before you submit to Gradescope**. If it doesn't work on your computer, *it will not work on Gradescope*.

CS 300 Assignment Requirements

You are responsible for following the requirements listed on both of these pages on all CS 300 assignments, whether you've read them recently or not. Take a moment to review them if it's been a while:

- Academic Conduct Expectations and Advice, which addresses such questions as:
  - How much can you talk to your classmates?
  - How much can you look up on the internet?
  - What do I do about hardware problems?
  - and more!

- Course Style Guide, which addresses such questions as:
  - What should my source code look like?
  - How much should I comment?
  - and more!

Getting Started

1. Create a new project in Eclipse, called something like **P10 Course Registration**.
    a. Ensure this project uses Java 17. Select “JavaSE-17” under “Use an execution environment JRE” in the New Java Project dialog box.
    b. Do **not** create a project-specific package; use the default package.

2. **Download** the **six** (6) starter files from the assignment page and add them to your project:
    a. PriorityQueueADT.java (the priority queue interface; generic)
    b. CourseRegTester.java
    c. CourseReg.java
    d. Course.java
    e. CourseQueue.java
    f. CourseIterator.java

You can begin by filling out the methods in the Course class – this is a relatively simple object. Make sure it is Comparable to other Courses, and that it contains the associated method.

Next, continue coding with the CourseRegTester class. Everything is laid out for you there, and it will direct you step-by-step on what to do next.

In a move that will infuriate many of you, the immediate tests for the autograder this week will be entirely focused on your tester class. We’ll try to help you determine whether your tests are sufficiently robust to help you test your OWN code, too, so that if your implementation passes all of your own tests, you should get nearly a perfect score on those hidden 30 points.

Assignment Submission

Once you’re satisfied with your work, both in terms of adherence to this specification and the academic conduct and style guide requirements, submit ALL your source code through Gradescope.

Your score for this assignment will be based on the submission marked “**active**” prior to the deadline. You may select which submission to mark active at any time, but by default this will be your most recent submission.

Copyright Notice

This assignment specification is the intellectual property of Mouna Kacem, Hobbes LeGault, Jeff Nyhoff and the University of Wisconsin–Madison and may not be shared without express, written permission.

Additionally, students are not permitted to share source code for their CS 300 projects on any public site.