Course Scheduler Final Project Part 2

This part of the Final Project is a continuation of Part 1 and will add additional functions for both the Admin User and the Student User.

This assignment is the last half of the final project and will be submitted as Final Project Part 2.

This phase of the project will implement the following additional Admin functions:

Display Course List of Students

Display all the students who are scheduled or waitlisted for the specified course for the current semester. The scheduled students should be displayed first and then the waitlisted students in waitlist order.

Drop Student

Remove the student from the list of students and remove them from any courses they are scheduled for in all semesters. For each course they were scheduled in, schedule the first student on the waitlist into the course. Display all of the changes that are made because of dropping the student.

Drop Course

Remove the specified course from the current semester. Remove all students scheduled and waitlisted for the course from the Schedule Table. Display all of the changes that are made because of dropping the course.

This phase of the project will implement the following additional Student function:

Drop Course

Remove the specified course from the student's schedule for the current semester. The student may be scheduled for the course or on the waitlist. Schedule the first student waiting for the course into the course. Display all of the changes that are made because of dropping the course.

Testing scenario:

A testing scenario will be provided to assist you in testing this application. It will be called Final Project Part 2 Test Script in Canvas.

Database considerations:

Your database will be identical to your database from Part 1. All of the tables should be empty when your project is submitted.

GUI Guidelines:

The user should be required to enter only unknown data. Drop down lists of known data such as Student names, Course Codes, or Semesters should be displayed for the user to select. Combo Boxes should be used for the drop-down lists on the form. When information is requested to be displayed, e.g., for a Display command, all of the requested information must be displayed. When a command is performed, the results of that command should be displayed to the user on the same display without the user needing to use a Display function to see what was done.

Submission Guidelines:

Don't forget to submit your zipped PROJECT folder and your zipped DATABASE folder as you did for Part 1. Zip the ENTIRE database folder and the ENTIRE project folder and submit the two zipped files in the assignment under one submission.

Note:
Your project must be named as it was for Part 1. The database must be named as it was for Part 1. All tables should be empty.

Grading Criteria:

In this project I will be looking for good OO design practices and this includes:

- Use of getter and setter methods for class variables

- Good naming of your classes, methods and variables

- Correct use of static and non-static methods

- The way you split this project into classes.

- All of your updates to the database must be done using SQL statements, do not use ResultSetTableModels to update the database.

- **If a SQL statement to update the database needs to contain a variable, then you must use PreparedStatements, do not use concatenation of strings to create the SQL statement.**

The Grading Rubric for Final Project Part 2 is posted in Canvas.

Note: Make sure you look at all the videos about this assignment and the Course Scheduler Design Layout in Canvas before starting this assignment.