Course Scheduler Final Project Part 1 Grading Rubric

Late policy of 4% per day late and using good object-oriented coding techniques still apply. Copying code from the internet does not count as writing your own code. Do not do this. The way to check your code is to use the Course Scheduler Final Project Part 1 test script provided. The Course Scheduler Final Project Part 1 test script will be used to test your assignment.

There are seven functions to Part 1 of the final project:
The Admin functions of: adding a semester, adding a course, adding a class, and adding a student and the Student functions of: displaying the classes available, scheduling a class, and displaying the class schedule for a student.

Preferably, the code you submit will be only for part 1, but if your code includes functionality for the second part of the project, that's ok too; we will only grade for the functionality for part 1 (so you need to make sure all parts for part 1 still work).

Database and project considerations:

- When you submit your project, you MUST submit both your zipped project folder and your zipped database folder (your grade will be 0 if you fail to submit your database when you submit your code. We need it to grade your assignment.)
- Your project code must be based on the template project provided for the current semester. If not, your grade will be 0.
- If your database does not work with your code, e.g., wrong table or column names, your grade will be 0.
- Your database should have the username and password of 'java' and 'java' as per the assignment and have the proper database name as per the assignment. Your project must also have the proper name as specified in the assignment.
- Your database must have the tables defined but the tables must be empty (-5 for not following these directions)
- You should have the following tables in your database with the appropriate columns: semester, courses, classes, students, and schedule. (-10 for each table)
- You should also have the appropriate primary keys.
- You must use SQL Prepare Statements (-10) when modifying the database.

Functionality:

- Correctly add a semester to the database (-10)
- Correctly add a course to the database (-10)
- Correctly add a class to the database (-10)
- Correctly add a student to the database (-10)
- Correctly display the available classes for a semester. (-10)
  - Display should be in columns and display all pertinent data.
- Correctly schedule a class for a student or waitlist the request (-10)

    - If the number of seats in the class have been filled by other students, the student is placed in a waitlisted status. (-5) for incorrect status.
- Correctly displaying the class schedule for a student for the current semester. (-10)
    - The classes scheduled and whether they are scheduled or waitlisted must be displayed.
- The priority order of a class waitlist must be maintained (-5)
    - It is strongly recommended that you use Timestamp to keep track of a student's position on the waitlist. This will come into play in Part 2.

GUI:

- Although the tab layout is recommended, it is not required. If you designed your GUI using a different type of layout, that's totally fine – just make sure you have all the required pieces of functionality and show all the appropriate information.
- Users entering data in textfields is only permitted for the Add functions. All other functions must use combo boxes to display data from the database for the User to select. (-5)
- The combo boxes should be updated automatically when additional data is added to the database (-3)
- When a command is performed, the results of that command should be displayed to the user without the user needing to perform any display function to see what was done. For example, if you add a student to the database, you should display the name of the student that was added. Likewise, after performing a Schedule Class function, you should display whether the student was scheduled in the class or placed in waitlist status. (-3)
- How you want to display information is up to you, but it must include all necessary information and be readable by the grader.