**Final Project Grading Rubric Room Scheduler**

Point deductions are given in **percentages.**

Late policy of 4% per date late and using good object-oriented coding still apply.

Copying code from the internet does not count as writing your own code. Do not do this.

A good way to check your code is to use the test script provided.

Because the second part of this project is dependent on the first part, the point deductions from part 1 will apply in addition to what is listed here.

There are five parts of this part of the project:

Adding a new room, canceling a reservation for a specific faculty and date (we assume each faculty will at most reserve one room per date), adding a new date, checking the status for a particular faculty, and dropping a room.

Because the last assignment must be graded very quickly for final grades, it is very important that you submit the database and project as directed.

Database and project considerations:

* When you submit your project, you MUST submit both your code and your database (-5 if you fail to submit your database when you submit your code. We need it to grade your assignment.)
* You need to close your project before you zip the database otherwise the graders will receive errors when they grade it.
* You must submit the zip of your entire Netbeans project instead of just the .java files this time
* Your database should have the username and password of ‘java’ and ‘java’ as per the directions
* ALL tables in your database must be EMPTY (-5 for not following these directions)
* Make sure you name your project folder and database with your name or user id

Functionality:

* Correctly adding a room to the database (-15)
  + Must include room name and number of seats
  + You may use number spinner or textfield to enter the number of seats
  + You must book any reservations you can from the waitlist.
* Correctly adding a new date to the database (-10)
  + The dates can be in Date or Varchar format in the database
  + If you accept strings for your dates, make sure you give the user directions on what format to enter
* Correctly checking status by faculty
  + Must display room and date faculty is reserved for (-3)
  + Must display reservations faculty are waitlisted for by date (-3)
  + Must have a dropdown list of faculties to choose from (-3)
* Drop room
  + Remove room from database (-3)
  + Faculty with this room reserved must get a reservation for another room for the date(s) they were originally booked, if possible. If not possible, the reservation should be added to the waitlist (-9)
  + Faculty should get reservations based on priority (their timestamp) (-2)
  + User should be informed of all actions (see below) (-3)
* Cancel Reservation (cancel faculty date)
  + Remove reservation or waitlist corresponding to that faculty and date from the database (-3 for each reservation and waitlist)
  + Must cancel reservation by faculty and date only (should not include room number) (-3)
  + If a reservation was cancelled, check the waitlist and make a reservation for any faculty waiting for that date that will fit in the room (-5)
  + Faculty from waitlist must be booked in order of their position (timestamp) (-3)

GUI:

* You can have a separate tab for each function, or you can combine multiple functions on the same tab. The big thing is that you have all pieces of functionality.
* You should have combo boxes to check faculty status, drop room, cancel faculty date, etc. (-2 for each combo box missing)
* 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 check the status to see what was done. For example, if you add a faculty to the database you should display the name of the faculty that was added and if the faculty was added to the database successfully. Likewise, after adding a reservation, you should display which faculty was booked on which room on what date. If a room is full, you must display information that the faculty was added to the waitlist
  + You will need to show the status for every action
    - For Part 2, this means displaying to the user if a room or date was successfully added to the database (or was not successfully added); for cancel faculty date this means showing that the reservation was successfully canceled, then showing which person from the waitlist was able to be added to the room (if there is one); for drop room, this means displaying how faculty are rebooked or waitlisted if they can’t be rebooked
  + When showing the results of a command, you should display the results of that particular command, not all of the contents for the database or that table in the database