

CPI221 - JDBC

50 points

Topics

- Object Oriented Programming
- Properties and methods
- SQL
- JDBC

Introduction

This is a very simple introduction to using JDBC and SQL in Java. This database is not intended to be “professional,” it is intended to allow the student to get comfortable with crafting and using queries and interacting with Java OOP.

Recommended Readings:

- <https://www.tutorialspoint.com/jdbc/>
- <https://www.codecademy.com/articles/sql-commands>

Use the following Guidelines:

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- User upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.

Important Note:

All submitted assignments must begin with the descriptive comment block. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment.

Programming Assignment

Description:

I have provided for you a database of made-up students with two tables. The first table is the students in the class, and the second tables are the grades for three assignments.

Student Data:

first name, last name, student ID (primary key)

```
CREATE TABLE `studentTable` (  
  `first_name` VARCHAR(20),  
  `last_name` VARCHAR(20),  
  `student_id` char(9)  
);
```

Assignment Data:

assignment number, student ID, max points, earned points, letter grade

```
CREATE TABLE `assnTable` (  
  `student_id` CHAR(9),  
  `assn_num` SMALLINT,  
  `max_points` SMALLINT,  
  `earned_points` SMALLINT,  
  `letter_grade` CHAR(1)  
);
```

Part 1 – JDBC and Queries (20 points of specifications)

Write a Java Console program to do the following:

Provide a menu and appropriate input to run the following queries:

1. Run a Query to display all of the students on separate lines
2. Run a query to display all of the students and their grades in blocks like this:

```
Lastname, Firstname:  
<Assn>: <points>/<max>  
... repeat for each assignment
```

3. Run a query for a specific student and display their results in the same format as #2
 - a. This query should ask the user to search via name or student ID
4. Run a query that displays results like #2 that uses the SQL LIKE statement to match last names that start with a certain sequence. Show results for all students that match the LIKE
5. Calculate the class average for a specific assignment
 - a. Prompt the user for the assignment number and calculate the average

Part 2 – Getting OOP (5 points of specifications)

Expand on the above program by creating a student and assignment class. Provide a menu option to calculate and show final grades.

The student should “has a” multiple assignments (an array or arrayList will do)

Load the data in the database into a collection of student objects.

Use those student objects to calculate the each individual student’s grades in the class.

Bonus +5

Use Java to create a final grades table that contains the student_id and the final grade in the database.

Grading of Programming Assignment

The TA will grade your program following these steps:

- (1) Compile the code. If it does not compile, 20% of the points given will be deducted. For example, if there are 20 points possible, you will earn 16 points if the program fails to compile.
- (2) The TA will read your program and give points based on the points allocated to each component, the readability of your code (organization of the code and comments), logic, inclusion of the required functions, and correctness of the implementations of each function.

Rubric:

Criteria	Levels of Achievement						
	A	B	C	D	E	U	F
Specifications 👍 Weight 50.00%	100 % The program works and meets all of the specifications.	85 % The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.	75 % The program produces mostly correct results but does not display them correctly and/or missing some specifications	65 % The program produces partially correct results, display problems and/or missing specifications	35 % Program compiles and runs and attempts specifications, but several problems exist	20 % Code does not compile and run. Produces excessive incorrect results	0 % Code does not compile. Barely an attempt was made at specifications.
Code Quality 👍 Weight 20.00%	100 % Code is written clearly	85 % Code readability is less	75 % The code is readable only by someone who knows what it is supposed to be doing.	65 % Code is using single letter variables, poorly organized	35 % The code is poorly organized and very difficult to read.	20 % Code uses excessive single letter identifiers. Excessively poorly organized.	0 % Code is incomprehensible
Documentation 👍 Weight 15.00%	100 % Code is very well commented	85 % Commenting is simple but solid	75 % Commenting is severely lacking	65 % Bare minimum commenting	35 % Comments are poor	20 % Only the header comment exists identifying the student.	0 % Non existent
Efficiency 👍 Weight 15.00%	100 % The code is extremely efficient without sacrificing readability and understanding.	85 % The code is fairly efficient without sacrificing readability and understanding.	75 % The code is brute force but concise.	65 % The code is brute force and unnecessarily long.	35 % The code is huge and appears to be patched together	20 % The code has created very poor runtimes for much simpler faster algorithms.	0 % Code is incomprehensible

What to Submit?

You are required to submit your solutions in a compressed format (.zip). Zip all files into a single zip file. Make sure your compressed file is labeled correctly - lastname_firstname3.zip.

The compressed file MUST contain the following:

```
<lastname>_<firstname>_hw3.c
```

No other files should be in the compressed folder.

If multiple submissions are made, the most recent submission will be graded, even if the assignment is submitted late.

Where to Submit?

All submissions must be electronically submitted to the respected homework link in the course web page where you downloaded the assignment.

Academic Integrity and Honor Code.

You are encouraged to cooperate in study group on learning the course materials. However, you may not cooperate on preparing the individual assignments. Anything that you turn in must be your own work: You must write up your own solution with your own understanding. If you use an idea that is found in a book or from other sources, or that was developed by someone else or jointly with some group, make sure you acknowledge the source and/or the names of the persons in the write-up for each problem. When you help your peers, you should never show your work to them. All assignment questions must be asked in the course discussion board. Asking assignment questions or making your assignment available in the public websites before the assignment due will be considered cheating.

*The instructor and the TA will **CAREFULLY** check any possible proliferation or plagiarism. We will use the document/program comparison tools like MOSS (Measure Of Software Similarity: <http://moss.stanford.edu/>) to check any assignment that you submitted for grading. The Ira A. Fulton Schools of Engineering expect all students to adhere to ASU's policy on Academic Dishonesty. These policies can be found in the Code of Student Conduct:*

[http://www.asu.edu/studentaffairs/studentlife/judicial/academic_integrity.h
tm](http://www.asu.edu/studentaffairs/studentlife/judicial/academic_integrity.htm)

ALL cases of cheating or plagiarism will be handed to the Dean's office. Penalties include a failing grade in the class, a note on your official transcript that shows you were punished for cheating, suspension, expulsion and revocation of already awarded degrees.
