CIS355A Week 6 Lab—Database Connectivity

**OBJECTIVES**

* Programmatic access to a MySQL database to add and display records

**PROBLEM: Student Management System**

A teacher needs the ability to store and retrieve student data. This includes

* student name;
* three test scores;
* average; and
* letter grade.

**FUNCTIONAL REQUIREMENTS**

You can code the GUI by hand or use NetBeans GUI builder interface.

Create a GUI which allows for input and display of student data.

It should include buttons to save a record, display all records.

Create a database and table to store student name and three test scores. (Note that average and grade are calculated by app.)

Student class

Create a Student class to manage the student data. It should have private instance variables of

* student name; and
* three test scores.

The class must have the following methods.

* A default and parameterized constructor
* Sets/gets for all instance variables
* A get method to calculate and return the average
* A get method to calculate and return the letter grade
* toString to display the name of the student

StudentDB class

Create a StudentDB class that is used to create a connection and interface with the database.

This class should have two methods.

* getAll—reads data from database, returns data in an arraylist of student objects
* add—writes a record to the database

GUI class

Insert button will take the info from the GUI (student name and three test scores) and insert a record into the table. Input should be cleared from the textboxes.

Display button will read the data from the database and creates a report in Console window, sample format below.

Name Test1 Test2 Test3 Avg Grade

Bruce Wayne 90 95 98 94.3 A

Clark Kent 65 70 90 75.0 C

**Sample GUI**



**RUBRIC**

|  |  |
| --- | --- |
| Student class * Has all required functionality
 | 10 |
| GUI class* Student record can be saved
* All student data can be displayed
 | 15 |
| StudentDB class* add method inserts a record into db.
* get method reads all records and returns in arraylist.
 | 15 |
| Code style | 5 |
| Lab Report | 10 |
| TOTAL | 55 |

**CODE STYLE REQUIREMENTS**

* Include meaningful comments throughout your code.
* Use meaningful names for variables.
* Code must be properly indented.
* Include a comment header at beginning of each file, example below.

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
Program Name: ProgramName.java
Programmer's Name: Student Name
Program Description: Describe here what this program will do
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**DELIVERABLES**

Submit as a SINGLE zip folder

* all java files; and
* the Lab report.

Follow assignment specification regarding class/method names.

Note that your Java file name must match class name (DO NOT rename).