**CMSC 214 Project 10**

**Task for Project 10(Covers Chapter 35 Java Advanced Database Programming)**

**Project Objective:**

* Effectively use an Advanced Database Programming
* Write a program that meets the following requirements(see below)

**Instructions:**

In addition to the goals of being able to write effective Java programs.

**Deliverables:**

In this programming project the student will design, develop, test and document a Java application.

**Projects submitted with evidence of plagiarism will be given a score of 0.**

**Description of a project (Batch update)**:

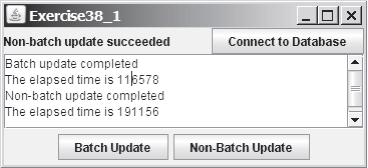
(Write a program that inserts a thousand records to a database, and compare the performance with and without batch updates, as shown in Figure 1(a). Suppose the table is defined as follows:

**create table Temp(num1 double, num2 double, num3 double);**

You can use any relational database: MySQL, Access or Oracle

Use the Math.random() method to generate random numbers for each record.

Create a dialog box that contains DBConnectionPanel, discussed in Exercise 32.3. Use this dialog box to connect to the database. When you click the Connect to Database button in Figure 1(a), the dialog box in Figure 1(b) is displayed.

****

35\_1

Figure 1a.

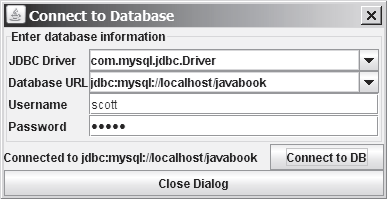


Figure 1b.

**Code** **Documentation and Style Requirements**The documentation requirement for all programming projects is one block comment at the top of the program containing the course name, the project number, your name, the date and platform/compiler that you used to develop the project. In addition, there should be at least one comment for each class in the program describing what that class does. Additional comments should be provided as necessary to clarify the program.

Indentation must be consistent throughout the program. Variable and method names should be descriptive of the role of the variable or method. Single letter names should be avoided. All constants, except 0 and 1, should be named. Constant names should be all upper-case. Variable names should begin in lower-case, but subsequent words should be in title case (e.g., finalSpeed).

Separate compilation must be used in accordance with standard Java practice. Every class must be saved in a separate .java file

**Rubric for Project 10**

**Grading:**

This project has three parts, with **100 total points** possible.

  (40 points) Carefully declare all the variables with appropriate data types. Use comments effectively to make the program more readable. Implement the calculations correctly.

  (40 points) Compile and build the Java program. Test the program using the Test Plan as described above and make sure the program produces the desired answer.

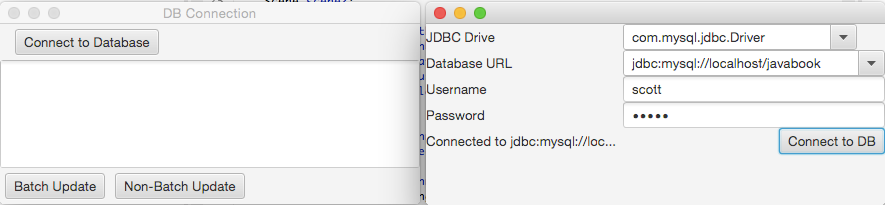
  (20 points) Submit your source code, screenshots, UML diagram, and algorithm as attached files to HW12 Folder

***Hints:***

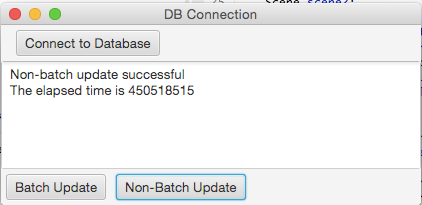
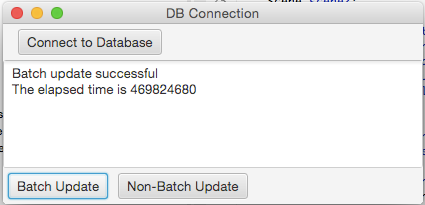
* ***Use Exercise32\_2java(see attachment)***
* ***Use Exercise32\_3.java(see attachment)***
* ***Use Exercise35\_2.java(see attachment)***
* ***Use Exercise35\_4.java(see attachment)***
* ***Use Listings 35.2 CopyFileToTable***
* ***You can use any database: MySQL, Access, Oracle***

***Instructions:***

1. Create a folder titled “lib” in the Java project. Add the .jar files from Blackboard into the folder.
2. Right click on the project name in the left sidebar in Eclipse. Go to Build Path > Configure Build Path.
3. Select Add External JARs on the right. When the new window appears select the JAR files in the lib folder. Click OK.
4. Run the program. Click Connect to Database. When the new window appears, select “com.mysql.jdbc.Driver” and “<jdbc.mysql://localhost/javabook>.” The username is scott and the password is tiger.
5. You can now use the buttons i the first window to execute the batch and non-batch updates.

******

***Screenshots:***

******

***UML:***

******