**CIS 357 Assignment 4**

**Topics: Thread, Network, Dababase Programming**

**Due:**noon, June 13th

**What To Submit:** All source files (.java), jar files, Javadoc generated files, and input/output files zipped to a single file called hw3-lastname.zip submitted to Canvas, a print out of Java file and output in class.

**Description:** Install MySQL on your computer (or lab computer). If you have a MySQL account on the csis machine, you can use it. Refer to the chap34 notes for installing MySQL, creating a database called scramble\_game, and a database table called result.

The following shows the overall architecture of the game:

Client Server

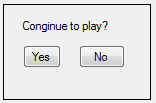
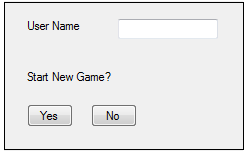
Database

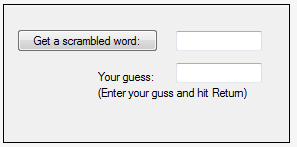
The client connects to the server via socket programming. (Using JSON is not allowed.) The server listens for the client connection, and keeps the connection until the client quits the game.

The server should be able to handle multiple clients playing the game at the same time. That is, the server should be multi-threaded. Assume that each client uses a different user name to avoid race condition.

Once the game is over, the result is saved back to the database. That is, the round/win/loss/guess data for the user is updated. At each end of the game for each player, display the current result and the overall result from all the players who already had played the game. (The result data should be sent back to the client machine and the client displays the result.)

You can use similar GUIs as the previous program, as seen. You can make them better – look better and play better. Refer to the previous assignment specifications for details of the requirements.





When the server connects to the database, use the following statement:

DriverManager.getConnection("jdbc:mysql://localhost/scramble\_game", "root", "");

Note: Do not reset the root’s password. If you use the MySQL on the csis machine, you can use the userid and password for that machine.

* Hint: Try to get simple things done first.
  + Database programming – see if the result is correctly saved and retrieved
  + Network programming – see if the data is correctly sent to the server and received from the server. Sending/receiving multiple data (like array or objects) may not be trivial, so you can try with simple data first.
* What to submit

1. Homework4-Client.java, Homework4-Server.java
2. ScramTest.java
3. Scramble.java
4. Results.java
5. scramwords.txt
6. results.txt
7. all files (html, css, …) generated by Javadoc.
8. Any additional java files.
9. A sample run results (screenshots) in hw3-out.doc (or docx).
10. client.jar and server.jar. These jar files should run the client program and server program once the jar files are double clicked.

Printout in class: All java files, input and output files, including the sample output.

This is for individual work! All codes should be your own!

* Bonus (5 points): If you show the working of your program in 3 different machines: for the client, the server, and the database.
* Bonus (20 points): Refer to <http://www.coolmath-games.com/0-hangman>. Create a hangman or hangwoman with five balloons and a monster. Each time the user’s guess is wrong, a balloon is popped (with the pop sound). After the user gets all words correct (in our example, four words), the hangman or hangwoman kicks the monster and walk away.