Project 1 – Unit 6

In this unit you have to provide Create, Update and Delete operations for persisting the LinkedHashMap into a Database.

The operation functions should be called whenever:

1. A new Automobile is added to the LinkedHashMap.
2. An Automobile is deleted from LinkedHashMap
3. An Automobile is updated.
4. An Automobile is deleted.

You will need to design and implement:

1. A database schema. (Be sure to apply rules of normalization discussed in class).
2. Create a set of classes for managing the Create, Update and Delete operations.
3. You should write a driver (in main) and test out the functionality in a console program on the server side.

Do’s and Dont;s

1. Implement Database functionality in the server.
2. Do not implement any functionality from the client side.
3. Do not save Option Choices.
4. Client interaction for creating, updating and deleting are not to be implemented or tested.Do' All functionality should be tested in a driver from Server side. In later versions (which will never be built in this course) the interactions with client can be setup.

How?

1. You will be given detailed examples in class on how to setup a database and connectivity.

Grading your Submission

1. Program Specification/Correctness (25 points)
a. No errors, program always works correctly and meets the specification(s).
b. The code could be reused as a whole or each routine could be reused.
c. Design is reusable and extensible.
d. Interfaces and abstract classes are applied for both unit 5 and 6.
e. DB schema is normalized.
f. Database classes are setup in a way so the SQL is read from a text file. (Improves modifiability).
g. Code is adequately tested and test runs are shown for both Unit 5 and Unit 6.

2. Readibility(5)
a. No errors, code is clean, understandable, and well-organized.
b. Code has been packaged and authored based on Java Coding Standards.

3. Documentation(5)
a. The documentation is well written and clearly explains what the code is accomplishing and how.
b. Detailed class diagram is provided.

4. Code Efficiency(10)
a. No errors, code uses the best approach in every case. The code is extremely efficient without sacrificing readability and understanding.

Reclaiming lost points

If you lost points in Units 1 through 4 you should:
1. Modify your project and fix issues that were reported to you (for which points were taken off).

2. Create a change log that shows:
a. What you were asked to change
b. How many points were taken off.
c. What changes you made in what files
d. Show test cases for changes made (if applicable)
e. How many points should be added back in your opinion.

3. TA's will review this change log, grade and return the lost points, if they feel that issue(s) have been corrected.

Lessons Learned

Things I learned that I would apply in 2nd Mini. – 50 points

Please create a well-organized list of design lessons learned from Project 1 and submit to cislabs04@gmail.com.

You will earn one point for each item. (Of Course content quality is important in this context).