SE 310 Software Architecture I

Homework Assignment 2 - Part C

We continue our assignment by expanding the requirements. This is a course about change, and how you respond to change determines how successful you are.

Instructions

Your task is to further develop a generic survey taking system by adding test taking functionality. A test is like a survey, but it has the addition of a correct answer. When you add a correct answer, you may NOT add it to the survey, question (and sub question) classes. A survey has no responsibility for a test and therefore shouldn't have any code regarding a test within it. You may refactor your survey for it to work better with test. You should make use of your existing survey for the majority of the functionality. You may also refactor certain functionality out of survey into a separate object that can be used by both survey and test as long as it doesn't have any test specific functionality. An example of this would be grading. A survey should have no visibility to grading functionality whatsoever. As before, the system is to be written in JAVA and built on your existing code base. The following system requirements must be met.

Develop a system that allows a survey **and a test** to be:

1. Created
2. Modified
3. Stored
4. Loaded
5. Taken
6. Displayed (with and without correct answers in the case of a test)
7. Tabulated
8. Graded (test only)

The survey / test should be stored in a file system, you must use serialization. Each survey / test definition should be stored in an individual file. Each individual's response to a survey / test should be stored in another individual file.

A survey **or a test** can be composed of any combination of questions in the form of:

1. True/False
2. Multiple Choice
3. Short Answer
4. Essay
5. Matching
6. Valid Date.

Each question can accept a single response, or if appropriate, multiple responses. Obviously, a True/False question would not accept multiple responses.

Each test question, aside from essays questions, should have some sort of correct answer. Each question, other than an essay, can therefore be graded.

A results module should be developed that indicates the totals for a particular survey or test.

For Part C, turn in a PDF or very high-resolution jpg/png of your UML generated from a computer program. Please DO NOT hand write your UML you will be given no credit if you do. It is ok to use a program that adds a watermark.

TIPS:

Past Questions/Answers

*** READ this section before asking your own questions!

1. Can a question, say Matching or Multiple choice with multiple responses, be partially right, and if so should partial credit be given for that question proportionate to the percentage correct?

   No, keep it simple. All right or all wrong.

2. Can different questions be weighted differently? E.g. A test has 10 questions; are all of them worth 10% each, or can the first 5 be worth 15% and the last 5 be worth 5%?

   No, keep it simple. All questions are weighted equally.

3. Because essays and short answers obviously cannot be graded automatically, should the final score when grading ignore the essays? E.g., I have a test with 10 questions and 2 additional essays, all 12 are weighted equally. The 10 questions can be graded automatically and are all right. Should the grade displayed be a 100% with a note about needing to grade the essays, or should the grade be an 83.3 with a note about needing to grade the essays?

   Short answers CAN be graded automatically. Don't grade the essays. Use 100% as the total possible score and do not grade the essays. Instead, leave a note in the output about the ungraded essay questions. So in this example, an 83.3 would be correct.

Late Policy

- Assignments submitted 1 hour to 1 week late will receive a 15% penalty.
- Assignments submitted 1 to 2 weeks late will receive an additional 10% penalty.
- Assignments submitted more than 2 weeks late will be subject to an additional 5% penalty for each week.