

CS 350 Software Design / SE 310 Software Architecture I

Homework Assignment - Part 1

NOTE: this is a very demanding project and it will require a lot of effort and a substantial amount of time. For part 1 of this project, spending a just few hours or submitting just the outline we will do in class will not be sufficient and will earn you a very low grade.

Instructions

Your task is to develop a generic survey/test taking system. The system is to be written in **JAVA**. The following system requirements must be met.

1. Develop a system that allows a survey/test to be entered, modified, stored, loaded, taken and tabulated/graded. The survey should be stored in a file system, you may use an XML or a binary format, but do not have to. Each survey should be stored in an individual file.
2. A survey/test can be composed of any combination of: True/False, Multiple Choice, Short Answer, Essay Answer, Matching, and Rank the choices. Each question can accept a single answer, or if appropriate, multiple answers. Obviously, a True/False question would not accept multiple answers.
3. Each test question, aside from essays, should have some sort of correct answer. This implies that each question can therefore be graded.
4. A results module should be developed that indicates the totals for a particular survey or test.

This assignment will be given in four parts.

For Part I, 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 important that you read the instructions for parts 2 and 3 of this project before you start designing your UML so you have complete picture of the scope of the project. You can find instruction for parts 2 and 3 in Bb Learn and we also will review them in lecture.

NOTE: If your code does not compile or does not work properly, you must indicate to the TA what works and what does not or you will be docked additional points. Also, note that if you cannot get portions of this program to function properly, you will not pass the course, regardless of your grades in the midterm and final exam. Successfully completing this assignment is a key part of this class.

TIPS:

Many students get themselves in trouble not because they do not code well, but because of their approach. This is a large, yet not complex, assignment for students at your level. With a good approach, you will save a lot of time. Therefore, please follow these hints to improve your chances of successfully completing the project on time.

1. Learn how to use serialization. It is usually easier than other file saving methods. Many students nod and wave when we mention this then code their programs and add the file saving later. When a bug arises, they do not know if it is in their main code or file saving/loading code. Instead, create a vector of objects; serialize it to a file, and then load it back into memory. If you understand the syntax and subtleties of serialization before you start, you will be far better off. Do this before you do your UML, so you understand what needs to be in your UML for file saving.
2. Students sometimes complain that when we give this assignment out we have thrown them in the deep end at the beginning of class. However, you don't have to feel like you are desperately trying to tread water. If this assignment was given and you were just asked to code, many of you would not think about it, code in a spaghetti manner, and just trudge through. Then like many projects, when you are done, you will say *I wish I had coded it this way or that way*. Now is your chance to think it through before you start. Still, it could be intimidating, so try to work on a slice of the program. Create a survey that just does true / false and multiple-choice questions. Allow it to create questions, display them, take them, save them, load them, etc. Learn from writing the smaller program. Then design your UML, make any changes to the original program (or throw it out altogether), and then code the program in its entirety.

Past Questions/Answers

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

1. When the user is under the survey mode, is he/she allowed to give a null answer, to any question, regardless of the type of question, whether it be multiple choice, short answer or matching? On the other hand, should the admin be given the option to make certain questions not require an answer? Or should the overall design allow for such a change to take place in the future? (I guess I am asking whether each question regardless of type has a default answer or not.)

No null answers are permitted.

2. Under the testing mode (or survey mode), is the user allowed to skip questions and return to them later? That is, should the user be allowed to return to previously answered questions before the final submission and modify any of his responses?

You are not allowed to skip questions.

3. Should the user be allowed to quit the test or survey before completion? If the user does decide to quit, should his previous responses be saved so that he can resume the test later by selecting from a bank of saved but incomplete tests?

You should not allow a survey to be quit in the middle.

4. If at all grading is a future requirement, should the design allow for such a change to be implemented down the line?

Yes, it should be a complete design.

5. Is the main class, the driver of the application, shown in the UML If so is it specified in a special way?

Yes, and not special way.

6. If one were to implement a Java interface, and have classes implement the interface, how is this represented in UML?

See the following link. They just add the word interface in the class:

- <http://www.informit.com/articles/article.aspx?p=29224&seqNum=2>
- <http://www.agilemodeling.com/style/classDiagram.htm>

7. Can a question, say Matching or Multiple-Multiple choice, 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.

8. 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.

9. 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% and a note about the essays.

10. I was just wondering if you wanted us to set up our classes as we did in class, and have each type of question have its own class. I was thinking about combining some of them. For example, making a WrittenAnswer class and giving it a max length attribute that defaults to no maximum, or giving the multiple choice class a choices attribute that defaults to True/False. Or is there something bad about this I'm not seeing?

You are free to design the assignment however you wish. Just keep in mind that you will be graded on your design.

11. Also, was it decided that more than one choice can be chosen for multiple choice questions, or was that just for ranking/matching?

Multiple choice, short answer, and essays can all have multiple answers.

12. Who decides the number of questions that should be there in the test/survey? Is it the admin?

Admin

13. How many T/F or MC or other types of questions should be present in both the survey and the test?

The admin of the test/survey decides how many questions are added to the survey.

14. Are we are supposed to make up the questions on our own?

The program allows the admin to make up any questions they wish.

15. And exactly what should we display as an output?

The program needs to be menu-based allowing you to access all the functionality discussed. You do not turn in output. You turn in a functional program.

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.