CS 350 Software Design / SE 310 Software Architecture I

Homework Assignment - Part 2: Creating, Displaying and Storing a Survey/Test to a File

Main driver 10%

Your program should operate from a text menu with options. For Part 2, you must have an option to create a new survey and add new question of types: true/false, multiple choice, short answer, essay, rank the choices, and matching.

i.e. Menu 1

1) Survey
2) Test

i.e Survey Menu 2	**i.e. Test Menu 2**
1) Create a new Survey	1) Create a new Test
2) Display a Survey	2) Display a Test
3) Load a Survey	3) Load a Test
4) Save a Survey	4) Save a Test
5) Quit	5) Quit

When option **1** is selected from **Menu 2**, then a follow up menu is shown.

i.e. Menu 3

1) Add a new T/F question
2) Add a new multiple choice question
3) Add a new short answer question
4) Add a new essay question
5) Add a new ranking question
6) Add a new matching question

Creating the Survey/Test: 27%

- True/False 3%
- Multiple Choice 3%
- Short Answer 3%
- Essay Answer 3%
- Rank the Choices 3%
- Matching 3%
- Handles improper input 3%
- Single answer per question 3%
- Multiple answers per each question 3%

When you enter a new question for a survey or test, you must ask for the appropriate information depending upon the type of question.

i.e **T/F** is selected from **Menu 3**

```
Enter the prompt for your True/False question:
User types their prompt here.
```

i.e **Multiple Choice** is selected from **Menu 3**

```
Enter the prompt or your multiple-choice question:
User types their prompt here.

Enter the number of choices for your multiple-choice question.
User types the number of choices.

Enter choice #1.
User types choice 1.

Enter choice #2
User types choice 2....
```

If you were filling out a test instead of a survey, then you would need to add an additional prompt and query the user for the correct answer.

i.e.

```
Enter correct choice
User types the number of the choice they want to be the correct answer
```

Reasonable error checking should be included. For example, the application should only allow a valid option to be entered.

Displaying a Survey/Test 16%

When option **2** is selected from **Menu 2**, the Survey/Test should be displayed to the screen. This requires that each question have a method to display itself.

- True/False 2%
- Multiple Choice 2%
- Short Answer 2%
- Essay Answer 2%
- Rank the Choices 2%
- Matching 2%
- Single answers per question 2%
- Multiple answers per question 2%

i.e. **Display a Survey** is selected from **Menu 2**

```
1) This is an example of a T/F question?
T/F

2) This is an example of a multiple-choice question with 3 choices?
A) Choice 1 B) Choice 2 C) Choice 3
```

etc...

i.e. **Display a Test** is selected from **Menu 2**

```
1) This is an example of a T/F question?
T/F
The correct answer is T

2) This is an example of a multiple choice question with 3 choices?
A) Choice 1 B) Choice 2 C) Choice 3
The correct choice is A) Choice 1
```

etc...

Loading a Survey/Test: 16%

When option **3** is selected from **Menu 2**, the Survey/Test must be loaded from a file. You can decide on the file type. Be aware, some types are easier to verify that they work than others. i.e. Serializing vs. XML. You should present a menu of possible files to load and allow the user to select one of the files.

i.e.

```
Please select a file to load:
1) Survey 1
2) Survey 2
3) Test 1
4) Survey 3
```

- True/False 2%
- Multiple Choice 2%
- Short Answer 2%
- Essay Answer 2
- Rank the Choices 2%
- Matching 2%
- Single answers per question 2%
- Multiple answers per question 2%

Saving a Survey/Test: 16%

When option **4** is selected from **Menu 2**, the Survey/Test must be saved to a file. You can determine the file type. Be aware, some types are easier to verify that they work than others. i.e. serializing vs. XML.

- True/False 2%
- Multiple Choice 2%
- Short Answer 2%
- Essay Answer 2%
- Rank the Choices 2%
- Matching 2%
- Single answers per question 2%
- Multiple answers per question 2%

Comments and Overall Style: 6%

PLEASE NOTE: You must include the entire project and all its files. Place these files in a Zip not Tar file. Load a single file to Drexel Learn with a ReadMe file that explains any issues, what method you used for saving a file, and where the sample files are located.

Also, your sample files should have a relative address from the binary. **Do not use absolute paths** as they won't work when we are grading them.

YOU MUST have file saving and loading working to submit this assignment. If you are missing a type of question or multiple answers, you can turn it in and you will lose the points for those types of questions. **Make sure you create sample files with at least one of each type of question in it.**

Handling Improper Input (9%)

Your program should gracefully handle inappropriate input and NEVER crash.

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.