SE 310 Software Architecture I

Homework Assignment 2 - Part B: Creating, Display, Loading, Storing, Taking, Modifying a Survey – 200 points

Main driver (20 points)

Your program should operate from a text menu. The menu must have options to create, modify, store, load, take and display a survey. A survey must have a combination of the following types of questions: true/false, multiple choice, short answer, essay, matching, and valid date.

e.g. Menu 1

1) Create a new Survey
2) Display an existing Survey
3) Load an existing Survey
4) Save the current Survey
5) Take the current Survey
6) Modifying the current Survey
7) Quit

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

e.g. Menu 2

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 date question
6) Add a new matching question
7) Return to previous menu

Creating the Survey (36 points)

- True/False (4 points)
- Multiple Choice (4 points)
- Short Answer (4 points)
- Essay (4 points)
- Date (4 points)
- Matching (4 points)
- Handles improper input (4 points)
- Single response per question (4 points)
- Multiple response per each question (4 points)

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

e.g. **T/F** is selected from **Menu 2**

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

e.g. **Multiple Choice** is selected from **Menu 2**

```
Enter the prompt for 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....
```

e.g. **Date** is selected from **Menu 2**

```
Enter the prompt for your date question:
User types their prompt here...
```

Displaying a Survey (32 points)

When option **2** is selected from **Menu 1**, the currently loaded survey should be displayed to the screen. This requires that each question have a method to display itself.

- True/False (4 points)
- Multiple Choice (4 points)
- Short Answer (4 points)
- Essay (4 points)
- Date (4 points)
- Matching (4 points)
- Single response per question (4 points)
- Multiple responses per question (4 points)

If the user doesn’t have a survey loaded, display the following message:

```
You must have a survey loaded in order to display it.
```

Then return the user to Menu 1.

Otherwise display the survey.

e.g. **Display a Survey** is selected from **Menu 1**

```
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


3) What is today’s date?
A date should be entered in the following format: YYYY-MM-DD
```

For a valid date question, you can choose to accept it in any reasonable format you wish (that contains year, month, and date), but automatically display a line with each date question that illustrates the format. Do not make up something like stardate 😉.

… This can be followed by any number of questions

Loading a Survey (12 points)

When option **3** is selected from **Menu 1**, the Survey must be loaded from a file. You must use serialization. You **MUST** present a menu of possible files to load and allow the user to select one of the files.

- True/False (2 points)
- Multiple Choice (2 points)
- Short Answer (2 points)
- Essay (2 points)
- Date (2 points)
- Matching (2 points)

e.g. Loading a Survey is selected from Menu 1

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

Saving a Survey (12 points)

When option **4** is selected from **Menu 1**, the Survey must be saved to a file. You must use serialization.

- True/False (2 points)
- Multiple Choice (2 points)
- Short Answer (2 points)
- Essay (2 points)
- Date (2 points)
- Matching (2 points)

If the user doesn't have a survey loaded, display the following message:

You must have a survey loaded in order to save it.

Then return the user to Menu 1.

Otherwise display the survey.

Modifying an existing Survey (24 points)

Create your program so that each type of question can be modified not simply deleted and replaced.

- True/False (4 points)
- Multiple Choice (4 points)
- Short Answer (4 points)
- Essay (4 points)
- Date (4 points)
- Matching (4 points)

When you select **option 6** from **Menu 1**, if the user doesn't have a survey loaded, display the following message:

```
You must have a survey loaded in order to modify it.
```

Then return the user to Menu 1.

If the user has a survey loaded, ask what question the user wishes to modify. e.g.

```
What question do you wish to modify?
User enters existing question.
```

If the selected question is **multiple choice**, the program should first ask whether or not to modify the prompt and then ask which choices to change.
e.g.

```
 Display the prompt for the existing question

 Do you wish to modify the prompt?
 User response entered
```

If the response is *Yes*

```
 Display current prompt.

  Enter a new prompt:
  New prompt entered
 Do you wish to modify choices?
```

If the response is *Yes* display the choices

```
Which choice do you want to modify?

A) Choice 1      B) Choice 2       C) Choice 3

Choice is entered

New Value is entered
```

Filling out a Survey (28 points)

After selecting **option 5 from Menu 1**, if the user doesn't have a survey loaded, display the following message:

```
You must have a survey loaded in order to take it.
```

Then return the user to Menu 1.

If the user has a survey, then the survey starts asking questions like:

e.g.

```
Enter the name of the survey you wish to take:
surveyName

1. What is your favorite movie?
Fletch

2. What is the most evil team in football?
A)   Dallas Cowboys    B) New England Patriots       C) New York Giants
A
```

Comments and Overall Style (10 points)

PLEASE NOTE: You must include the entire project and all its files. Place these files in a Zip file, not Tar file. Upload and submit a single file to Drexel Bb Learn with a ReadMe file that explains any issues, and where the sample files are located.

Also, your sample files should have a relative address to the serialized files. **Do not use absolute paths** as they won't work when we are grading them. **Do not hard code file separators.** A review of how to handle files properly is in the JAVA review.

YOU MUST have file saving and loading working to submit this assignment. If you are missing a type of question or have not implemented the ability to accept multiple responses per question, you can turn it in, but 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.

Aspects to consider

Multiple Choice questions (multiple responses):

If the question allows multiple responses for a multiple choice question, you must to determine how many options the user should enter. Please make it obvious.
e.g.

```
What teams are evil in football? Please give two choices:
A) Dallas Cowboys       B) New England Patriots       C) Philadelphia Eagles

 A
 B
```

Multiple essay questions:

Some students question whether an essay question can have multiple responses. There is no problem with this. Each essay could have two paragraphs.

e.g.

```
Give two reasons why Star Wars is better than Star Trek:

 A) Star Wars has Yoda, Star Trek has Tasha Yar. Nuff said.
 B) Star Wars consistently breaks new ground with effects. Star Trek follows suit.
```

Matching questions:

You should make your Matching questions look presentable. i.e. two columns neatly formatted. This really is not difficult. Then you can enter a letter and number combination on each line.

```
 Match the team with the city

 A) Yankees       1) Philadelphia
 B) Phillies      2) New York
 C) Red Sox       3) Boston

 A 2
 B 1
 C 3
```

Saving the responses of a Survey (16 points)

After each survey is taken, the results must be stored in a file. You should store each user's response set in a separate file.

Handling Improper Input (10 points)

Your program should gracefully handle improper input and NEVER crash or quit unexpectedly.

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.**

Testing

Want to save yourself from accidently losing points? Upload your code somewhere else, like TUX and make sure it works there.

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.