SE 310 Software Architecture I

Homework Assignment 2 - Part D: Creating, Display, Loading, Storing, Taking, Modifying, Grading Test & Tabulating a Survey/Test – 100 points

Main driver (6 points)

Your program should operate from a text menu with options. For Part D, you must have an option to create a test with the following types of question: true/false, multiple choice, short answer, essay, matching and valid date.

e.g.
Menu 1

1) Survey
2) Test

Survey Menu 2

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) Modify the current Survey
7) Tabulate a survey
8) Return to previous menu

Test Menu 2

1) Create a new Test
2) Display an existing Test without correct answers
3) Display an existing Test with correct answers
4) Load an existing Test
5) Save the current Test
6) Take the current Test
7) Modify the current Test
8) Tabulate a Test
9) Grade a Test
10) Return to the previous menu

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

e.g.

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

Creating the Test: (18 points)

- True/False (2 points)
- Multiple Choice (2 points)
- Short Answer (2 points)
- Essay (2 points)
- Date (2 points)
- Matching (2 points)
- Handles improper input (2 points)
- Single response / answer per question (2 points)
- Multiple responses / answers per each question (2 points)

When you enter a new question for a test, you must ask for the appropriate information depending upon the type of question. This part should be a reuse a majority of survey code.

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

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

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

```
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 3**

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

When you are creating out a test instead of a survey, you would need to add an additional prompt and query the user for the correct answer.

e.g.

```
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 Test without correct answers (2 points)

This shouldn’t require any work.

Displaying a Test with correct answers (9 points)

When option **3** is selected from **Test Menu 2**, the Test should be displayed to the screen with correct answers. This requires that each question have a method to display itself.

- True/False (1 points)
- Multiple Choice (1 points)
- Short Answer (1 points)
- Essay Answer (1 points)
- Date (1 points)
- Matching (1 points)
- Single response / answer per question (1 points)
- Multiple responses / answers per question (1 points)

e.g.

```
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
```

Loading a Test (2 points)

When option **4** is selected from **Test Menu 2**, the Test must be loaded from a file. This is largely the same as surveys except you also have to load the correct answer.

e.g.

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

Saving a Test (2 points)

When **option 5** is selected from **Test Menu 2**, the Test must be saved to a file. This is largely the same as surveys except you also have to save the correct answer.

Filling out / Taking a Test (2 points)

After selecting **option 6** from **Test Menu 2**, the user is prompted to enter the name of the test to take and then the questions are asked and answered. There is no difference between taking a test and a survey. You should be able to reuse your code.

Modifying an existing Test (12 points)

After selecting **option 7** from **Test Menu 2**, modify your program so that each type of question can be modified not simply deleted and replaced. For this part of the assignment you should be able to reuse your existing code and just add code for modifying the correct answer, if one exists.

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

When you select **option 7** from **Test Menu 2**, if the user doesn't have a test loaded output the following message:

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

Then return the user to Menu 1.

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

```
What question do you wish to modify?
Enter 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 answer 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
```

Tabulate a Test/Survey (24 points)

Some questions can be tabulated, others can simply show the responses by users, placed together. If the user selects **option 7** from **Survey Menu 2** to or **option 8** from **Test Menu 2**, the following should happen**:**

For each test or survey tabulated**:**

- True / False, or Multiple Choice: Display the total number of responses for each choice.
- Short Answer, Date: Display the total number of each response. Do not worry about combining spelling issues and such.
- Essays: Simply list all of the answers
- Matching: List the number of each permutation submitted and the permutation.

Example: 5 responses given to a True/False Question:

Question:

```
Star Wars is the Greatest Movie Franchise. True / False
```

Responses:

```
True
True
False
True
True
```

Tabulation:

```
Star Wars is the Greatest Movie Franchise.
True: 4
False: 1
```

Example: 4 responses given to a to Multiple Choice Question:

Question:

```
What is your favorite Star Wars movie?
A) I          B) II       C) III      D) IV       E) V          F) VI
```

Replies:

```
D
E
D
E
```

Tabulation:

```
What is your favorite Star Wars movie?
A: 0
B: 0
C: 0
D: 2
E: 2
F: 0
```

Example: 5 responses given to a Short Answer Question:

Question:

```
What is your favorite Star Wars movie?
```

Responses:

```
IV
V
V
IV
V
```

Tabulation:

```
What is your favorite Star Wars movie?
IV 2
V 3
```

Example: 3 responses given to an Essay Question:

Question:

```
Why is your Star Wars your favorite movie?
```

Responses:

```
Star Wars combines a mythos second to none.

Could it get any better than Darth Vader being Luke Skywalkers' father?

Lucas created a completely new world of special effects.
```

Tabulation:

```
Why is your Star Wars your favorite movie?

Star Wars combines a mythos second to none.

Could it get any better than Darth Vader being Luke Skywalkers' father?

Lucas created a completely new world of special effects.
```

Example: 3 responses given to a Date Question:

Question:

```
When will the stock market rebound?
```

Responses:

```
5/1/2020
5/7/2021
9/1/2022
9/1/2022
```

Tabulation:

```
When will the stock market rebound?
9/1/2022
2

5/7/2021
1

5/1/2020
1
```

Example 3 responses given to a Matching Question:

Question:

```
Match the team with the city

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

Responses:

```
A 2
B 1
C 3

A 1
B 2
C 3

A 2
B 1
C 3
```

Tabulation:

```
Match the team with the city
A) Yankees          1) Philadelphia
B) Phillies         2) New York
C) Red Sox          3) Boston

2
A 2
B 1
C 3

1
A 1
B 2
C 3
```

Grading (18 points)

When you select **option 9** from **Test Menu 2**, a numerical grade must be generated for a test. Prompt the user as follows:

```
Select an existing test to grade:
   1) Test 1
   2) Test 2
   3) Test 3
```

The user selects an option and then the user is presented with a list of previously saved responses. The user then selects one to be graded:

```
Select an existing response set:
   1) Test 1 - Response 1
   2) Test 1 – Response 2
   3) Test 1 – Response 3
```

After the user makes a selection, a grade is computed and displayed:

```
You received an 80 on the test. The test was worth 100 points, but only 90 of
those points could be auto graded because there was one essay question.
```

Observe the output carefully. Make all tests worth 100 points. Divide all questions so they have equal weight. In this example, the user got 8 out of 9 auto gradable questions correct. There was one essay question. If essays are part of a test, remove them from the possible gradable points as shown in this example.

Comments and Overall Style (3 points)

PLEASE NOTE: You must include the entire project and all its files. Place these files in a Zip file, not Tar file. Upload an 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 from the binary. **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 answers 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.**

Saving the responses of a Test (2 points)

After each test is taken, the responses must be stored in a file. This functionality is the same as the one you implemented for saving responses to a Survey.

Handling Improper Input (0 points directly)

Your program should gracefully handle inappropriate input and NEVER crash. No specific points are allocated, but if input is not handled properly, points will be deducted from the section with the errors.

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.

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.