COMP 2710 – Project 4: LET'S TAKE A QUIZ

Points Possible: 100
Deadline: **11:59pm, Friday, April 4th, 2025 (Central Time)**

Goals:

- Able to learn how to use linked data structures.
- Able to perform basic operations on such structure.
- Able to understand programming requirements purely via outputs.
- Able to use conditional compilation to switch between debugging and production mode.
- Able to further enhance program testing skills with Unit testing and Integration Testing.
- Able to further enhance functional programming skills.

Caution:

Throughout this project, you are forbidden to use any data structures involving arrays, vectors or any library that can create a contiguous array of elements. Arrays are keyworded as double square bracket symbols in C++. You will receive an automatic zero if you define any of these.

SCROLL DOWN TO GET STARTED

PART A: PHASE 1 – LINKED QUESTION LIST

I. Description:

In Canvas, you can create quizzes and tests for students, allowing you to assess their performance in class. These quizzes are organized using a linked data structure, enabling students to navigate through questions or jump directly to a specific one. In this project, you'll investigate how it is done, and then develop a similar quizzing system as a C++ program.

In a basic sense, the program will present a question to the test taker, accept their response, and check it against the correct answer. If the student answers correctly, they'll move on to the next question; if not, the correct answer will be shown. Each question has a designated point value, and each quiz will consist of a specific set of questions with a total point maximum. Once the student completes the quiz, their final score will be displayed.

Note 1: You must provide the following user interface with relatively similar output structure. You do not need to color the texts that are highlighted in red in the sample output. Study this and try to understand what the program is doing. Also, replace the word "Li" with your name.

```
*** Welcome to Li's Testing Service ***

=== QUESTION 1 ===
Type of question [mcq/tf/wr]: mcq

Enter a question: Which of the following approach is used by C++?
[At any time, type 'quit()' to exit]

Enter choice A: Left-right
Enter choice B: Right-left
Enter choice C: Bottom-up
Enter choice D: Top-down
Enter choice E: quit()

Select correct answer: E
[Answer not recognized, please try again!]

Select correct answer: c
Enter point value: .ab
[Not a point value, please try again!]

Enter point value: 902ls
[Not a point value, please try again!]
```

```
Enter point value: 10.00
Question saved. Continue? [y/n]: yes
[Command not recognized, please try again!]

Question saved. Continue? [y/n]: y

=== QUESTION 2 ===
Type of question [mcq/tf/wr]: kjs82
[Command not recognized, please try again!]

Type of question [mcq/tf/wr]: tf

Enter a question: In a flowchart, the switch diamond contains a
condition requiring a true or false answer.

Select correct answer: A
[Answer not recognized, please try again!]

Select correct answer: false
Enter point value: -12.50
[Not a point value, please try again!]

Enter point value: 12.50
Question saved. Continue? [y/n]: y

=== QUESTION 3 ===
Type of question [mcq/tf/wr]: wr

Enter a question: By default, all the files in C++ are opened in what
mode?

Type correct answer: text
Enter point value: 17.50
Question saved. Continue? [y/n]: n

=== SESSION LOG ===
Total questions: 3
Total point values: 40

/!\ Begin assessment? [y/n]: y

Question 1: Which of the following approach is used by C++?
   A. Left-right
   B. Right-left
   C. Bottom-up
```

```
   D. Top-down

Your answer: C
[Your answer is correct!]

Question 2: In a flowchart, the switch diamond contains a condition
requiring a true or false answer.

Your answer [true/false]: C
[Answer not recognized, please try again!]

Your answer [true/false]: true
[Your answer is incorrect. The correct answer is false.]

Question 3: By default, all the files in C++ are opened in what mode.

Your answer: TEXT
[Your answer is correct!]

/!\ Assessment Complete.

=== SESSION LOG ===
Correct answers: 2/3
Final score: 27.5/40

*** Thank you for using the testing service. Goodbye! ***
```

Note 2: Except for when it is clearly stated to accept a different input than what is being requested, you must handle all errors caused by illegal inputs or illegal logic at any time by default. If you fail to catch any of these mistakes, you will lose a minimum of 15 points for **Part B**.

II. Question types:

There are three types of question you can use to assess a student. These questions award arbitrary point values, which are non-negative and are assigned upon user input.

- **Multiple choice question**: `[mcq]`

  This question type presents several answer options labeled A, B, C, D, E, etc. At least one option must be available. Students select the correct option by typing the corresponding letter. The input is case-insensitive, both during and outside of the assessment.

- **True/False**: `[tf]`

This question type offers only two answer choices: true or false. Students need to type one of these answers to earn points. This input is also case-insensitive in all contexts of assessment.

- **Written response**: `[wr]`

  This type requires students to type an exact response relevant to the question's context. While the input is also case-insensitive, it must match letter for letter precisely.

III. Linkin' the questions:

Since the total number of questions is unknown at the beginning and must be generated dynamically, a linked list is an ideal data structure for this task. Feel free to choose whatever type of linked list that suits you. Looking for easy implementation? Choose singly linked list. Looking for greater efficiency? Choose doubly or circular linked list.

Now, consider the following questions:

1. What data and object need to be stored in each node of this linked list of questions?
2. Given the types of questions available, can you determine the appropriate type of data to store for each?
3. True/false and written response questions are pretty straightforward, but how will you handle multiple-choice questions? Is there any way you can dynamically create additional answer choices within a linked list node, and without relying on an array?

Answer yourself those question and experiment on a simple linked list first. Then move on to create a suitable structure for the MCQs and incorporate it into your original list.

Note 3: Functional programming is a must in this project. You can create functions that initialize the linked list, add a new question node to the list, write input data to each node, and/or a function to add answer choice for the multiple-choice question. It's all up to you to decide.

III. Let's take a quiz:

After you are done creating a question bank, you will move on to trying out the quiz and you should prompt whether to continue or not at this point.

Now, from the students' perspective, they will look at the question and follow the directions. Your program needs to compare their input with the actual answer on file. If they match, the student is awarded the question points, else the program should display the correct answer.

When the student is finished, display the relevant statistics of the assessment.

PART A: PHASE 2 – OPERATIONS ON LINKED QUESTION LIST

I. Description:

You now have yourself a basic quizzing system, but it is still far from being a practical system. Given that there is no way for you as the test creator to move back and forth to edit or delete questions, and for the students as test takers to change their answer choices or skip those questions that are giving them a hard time and come back later to do it. Within this phase, you will implement such functionalities in order to make your testing system more robust.

II. Operations during test creation:

Note 4: You should study the following user interface and must have yours in a relatively similar structure, which has been updated from Phase 1 to accommodate operations on linked list.

```
*** Welcome to Li's Testing Service ***

Do you want to?
   1. Create new question.
   2. Edit question.
   3. Delete question.
   4. Finish.
Select an action: 1

=== QUESTION 1 ===
Type of question [mcq/tf/wr]: mcq

Enter a question: Which of the following approach is used by C++?
[At any time, type 'quit()' to exit]

Enter choice A: Left-right
Enter choice B: Right-left
Enter choice C: Bottom-up
Enter choice D: Top-down
Enter choice E: quit()

Select correct answer: C
Enter point value: 10.00
Question saved.

Do you want to?
```

```
   1. Create new question.
   2. Edit question.
   3. Delete question.
   4. Finish.
Select an action: 1

=== QUESTION 2 ===
Type of question [mcq/tf/wr]: kjs82
[Command not recognized, please try again!]

Type of question [mcq/tf/wr]: tf

Enter a question: In a flowchart, the switch diamond contains a
condition requiring a true or false answer.

Select correct answer: false
Enter point value: 12.50
Question saved.

Do you want to?
   1. Create new question.
   2. Edit question.
   3. Delete question.
   4. Finish.
Select an action: 1

=== QUESTION 3 ===
Type of question [mcq/tf/wr]: wr

Enter a question: By default, all the files in C++ are opened in what
mode?

Type correct answer: text
Enter point value: 17.50
Question saved.

Do you want to?
   1. Create new question.
   2. Edit question.
   3. Delete question.
   4. Finish.
Select an action: 1

=== QUESTION 4 ===
Type of question [mcq/tf/wr]: mcq
```

```
Enter a question: Which of the following is a correct identifier in C++?
[At any time, type 'quit()' to exit]

Enter choice A: VAR_1234
Enter choice B: $var_name
Enter choice C: 7VARNAME
Enter choice D: 7var_name
Enter choice E: quit()

Select correct answer: A
Enter point value: 10.00
Question saved.

Do you want to?
   1. Create new question.
   2. Edit question.
   3. Delete question.
   4. Finish.
Select an action: 2

Select a question to edit, or type quit() [1-4]: 0
[That question does not exist!]

Select a question to edit, or type quit() [1-4]: 1

================================
=== QUESTION 1 SAVED VALUES ===
================================
   1. Type: mcq
   2. Question: Which of the following approach is used by C++?
   3. Answer choices:
     A. Left-right
     B. Right-left
     C. Bottom-up
     D. Top-down
   4. Correct answer: C
================================
Type a number to edit, or type quit(): 3

Enter choice A: Pyramid approach
Enter choice B: Reversed pyramid approach
Enter choice C: Hourglass approach
Enter choice D: quit()
Question saved.
```

```
Type a number to edit, or type quit(): 2

Enter a question: Choose the correct coding approach for Python.
Question saved.

Type a number to edit, or type quit(): quit()

Select a question to edit, or type quit() [1-4]: quit()

Do you want to?
   1. Create new question.
   2. Edit question.
   3. Delete question.
   4. Finish.
Select an action: 3

Select a question to delete [1-4]: 2
Question 2 deleted.

Do you want to?
   1. Create new question.
   2. Edit question.
   3. Delete question.
   4. Finish.
Select an action: 1

=== QUESTION 4 ===
Type of question [mcq/tf/wr]: tf

Enter a question: In a flowchart, the switch diamond contains an
expression whose value determines which path is chosen.

Select correct answer: true
Enter point value: 2.50
Question saved.

Do you want to?
   1. Create new question.
   2. Edit question.
   3. Delete question.
   4. Finish.
Select an action: 4


=== SESSION LOG ===
```

```
Total questions: 4
Total point values: 40

/!\ Begin assessment? [y/n]:
```

Note 5: Again, all invalid inputs must be handled appropriately. For example, if you type in a non-existing question number or you select a wrong input, it should prompt for a retry.

There are 4 operations you will add to your Phase 1 code during the creation of questions:

- **Create a new question**:
  Instead of asking whether the user wants to continue, you put this option in a list. A user will need to type in the number to create a new question. At least 1 question is needed.

- **Edit an existing question**:
  This option enables the user to edit any existing question node. Depending on the question type, you will provide specific editable options. For multiple-choice questions, refer to the output above which displays the current information in the node, allowing the user to select a number to modify a particular data value. For others, feel free to be creative while following the MCQ example and have your own unique approach.

- **Delete an existing question**:
  This option enables a user to delete a selected question on file. Once a question is deleted, the question nodes before and after it will be linked together.

  Note 6: Be absolutely careful when deleting a node from a linked list. If you make mistakes when reestablishing the links of the nodes, or forget to free up memory, you could run into errors such as segmentation fault or memory leaks.

- **Finish creating the question bank**:
  This option simply ends the question creation process and prompt the user if they want to continue to the assessment phase.

III. Operations during assessment:

Note 7: You should study the following user interface and must have yours in a relatively similar structure, which is a continuation to the assessment phase.

```
/!\ Begin assessment? [y/n]: y
```

```
Do you want to?
   1. Go to next question.
   2. Jump to question.
   3. Submit.
Select an action: 2

Jump to question [1-4]: 3

Question 3: Which of the following is a correct identifier in C++?
   A. VAR_1234
   B. $var_name
   C. 7VARNAME
   D. 7var_name
Your answer: a

Do you want to?
   1. Go to next question.
   2. Jump to question.
   3. Submit.
Select an action: 1

Question 4: In a flowchart, the switch diamond contains an expression
whose value determines which path is chosen.

Your answer [true/false]: true

Do you want to?
   1. Go to next question.
   2. Jump to question.
   3. Submit.
Select an action: 1

Question 1: Choose the correct coding approach for Python.
   A. Pyramid approach
   B. Reversed pyramid approach
   C. Hourglass approach

Your answer: C

Do you want to?
   1. Go to next question.
   2. Jump to question.
   3. Submit.
Select an action: 2
```

```
Jump to question [1-4]: 4

Question 4: In a flowchart, the switch diamond contains an expression
whose value determines which path is chosen.

You answered: true

Do you want to?
   1. Edit this answer.
   2. Go to next question
   3. Jump to question.
   4. Submit.
Select an action: 2

Question 2: By default, all the files in C++ are opened in what mode.

Your answer: TEXT

All questions answered. Do you want to?
   1. Go to next question.
   2. Jump to question.
   3. Submit.
Select an action: 1

Question 3: Which of the following is a correct identifier in C++?
   A. VAR_1234
   B. $var_name
   C. 7VARNAME
   D. 7var_name
You answered: a

Do you want to?
   1. Edit this answer.
   2. Go to next question
   3. Jump to question.
   4. Submit.
Select an action: 1

Your new answer: B

All questions answered. Do you want to?
   1. Go to next question.
   2. Jump to question.
   3. Submit.
Select an action: 3
```

```
/!\ Assessment Complete.

=== SESSION LOG ===
Correct answers: 3/4
    Question 1: C
    Your answer: C

    Question 2: text
    Your answer: TEXT

    Question 3: A
    Your answer: B

    Question 4: true
    Your answer: true

Final score: 30/40

*** Thank you for using the testing service. Goodbye! ***
```

Note 8: Notice this time, you don't show the correct answers immediately after answering it because you are allowing the students to go back and fix their answer. You should only reveal it when the student has submitted their test.

There are 3 operations you will need to add to your Phase 1 code during the assessment phase:

- **Go to the next question**:

  This option works exactly as described. At the beginning of the test, the next question is the first question. If a student skips around and completes some questions, the next question will be the one that hasn't been answered yet, based on the order of access in the linked list.

  For instance, say there are 7 questions in sequence (1 → 2 → 3 → 4 → 5 → 6 → 7) and the student jumps to question 3 and finishes questions 3 and 4, then jumps to 7 and completes it. If they return to question 2 and choose to go to the next question, the system will direct them to question 5, followed by 6, then 1.

  If there are no remaining unanswered questions, it will proceed to the next question in the usual order. However, it will display prompts similar to the jump method below. This is to ensure that students have the opportunity to make any changes they wish.

- **Jump to a question**:

Go to any of the question number and let the student answer the question. If already answered, provide a prompt asking whether the students want to edit their answer. In that prompt, it is advised to show what the question was and what they have answered so that they know if changing their answer is a good idea. Do this by yourself, apply what you know from the question creation process above.

- **Submit the test**:

  Submit the test and provide the correct answers and relevant statistics. There should be a prompt pops up to warn the student if they have unanswered questions. If they intent to proceed, unanswered questions will be marked as zero.

PART B: TESTING YOUR PROGRAM

You have learned to use Unit Testing in Project 2 to test each of your program functions using `assert()` and use Integration Testing in Project 3 to evaluate your program as a complete unit. For this project, you will attempt to combine both methods to strengthen your testing skills.

I. Unit testing:

Write a test driver to perform unit testing on all functions of your linked question list. Remember that unit testing uses `assert()` to verify whether the function or a logic behaves as you would expect. You are free to choose whatever type of assertion you want to do as long as you put a comment on what you are testing clearly in your code and have your assertion in a function.

Note 9: You will not receive credits for this section if you fail to do them. We expect there should be a comment block on top of your testdriver function explaining what you are testing.

For this task, you are required to make at least 5 assertions to verify your program execution. Here is a list of suggested testcases you can put your testing skills to practice:

1. Test whether you can have zero question in your test bank and the program will behave as expected.
2. Test whether you can delete all questions after setting them up. This involves checking if the linked list is empty at the point of the last delete.
3. Test whether a student can submit the test without answering any question.
4. Test a simple hard-coded flow of operation: you will program a function that automatically add in the questions, point values and relevant data points, then proceed to the assessment as a role of a dummy test student and hard code the answer. Verify that the final statistics match what you expect.

This list is not exhausted 'cause there are millions of ways you can do this. Make sure the testing procedure is logical and work as normal.

II. Integration testing:

You will do this task by thinking of all possible inputs to the program and verify them. There is no need for you to create files and read them into the program. You only need to make sure that your program can handle illegal inputs of any type, malicious behavior related to linked list such as segmentation fault or memory leak, and checking for undeleted node pointers.

III. Switching between debug and production:

As you work, it may become challenging to distinguish between the testing and the actual running portion of your code, so conditional compilation will be your solution. You'll need learn how to switch between the debug and the production version of your code using preprocessor switches.

Once you have written all testdrivers in **Part B**, **Section 1 + 2**, you can move them into part of the preprocessor directive `#ifdef` to keep them clear from the regular version. The following code block, which lies on top of your `.cpp` file, illustrates where you should put them:

```
#define UNIT_TESTING
#ifdef UNIT_TESTING
//add your unit testing code here
#else
//add your code for the product version here
#endif
```

In your program, the portion of code that is compiled depends on whether a preprocessor macro by that name is defined or not. For example, if the `UNIT_TESTING` macro has been defined, i.e., line is enabled, then your unit testing code portion is compiled, and the production code is ignored. If the macro is not defined, the reverse will happen.

These macros look a lot like `if` statements, but macros have completely different behavior. More specifically, an `if` statement decides which statements of your program must be executed at run time, while an `#ifdef` controls which lines of code in your program are actually compiled. The keyword here is compilation versus runtime.

IV. Sample debugging output:

Below is a sample output when you switch your code to debugging. You do not have to follow this exactly. However, at least have a clear output structure like the sample below when you do it yourself. Readability of your output will contribute to your grade.

```
***This is a debugging version ***

Unit Test Case 1: Ask no question. The program should give a warning
message.
Warning - the number of questions to be asked must equal to or be larger
than 1.
Case 1 Passed
```

```
Unit Test Case 2.1: Ask 1 question in the linked list. The tester enters
an incorrect answer.
Question: How long was the shortest war on record?
Answer: 85
Your answer is wrong. The correct answer is 38
Your total points 0

Case 2.1 passed

Unit Test Case 2.2: Ask 1 question in the linked list. The tester enters
a correct answer.
Question: How long was the shortest war on record? Answer: 38
Your answer is correct! You receive 100 points.
Your total points: 100

Case 2.2 passed

Unit Test Case 3: Ask all the questions of the last trivia in the linked
list. Question: How long was the shortest war on record?
Answer: 38
Your answer is correct! You receive 100 points. Your total points: 100

Question: What was Bank of America's original name? (Hint: Bank of Italy
or Bank of Germany)?
Answer: Bank of Germany
Your answer is wrong. The correct answer is Bank of Italy.
Your total points 100

show question here
Add your answer here
. . .
Case 3 passed

Unit Test Case 4: Ask 5 questions in the linked list. Warning - There is
only 3 questions in the list.
Case 4 passed

*** End of the Debugging Version ***
```

PART C: GRADING AND POLICIES

I. Grading:

100 points maximum, criteria ordered by parts, and from lowest to highest in point value.
All these criteria work in harmony, so if you miss one or two, you might be doing so on others.

PART A: (50 points), of which

FUNCTIONAL PROGRAMMING (10 points)

PHASE 1 (20 points, all carry equal point value).

1. You can use a linked structure to store the question data.
2. Your program can add multiple choice questions and use them during assessment.
3. Your program can add true/false questions and use them during assessment.
4. Your program can add written-response questions and use them during assessment.
5. You can perform the assessment by asking any type of question, record responses and compare them against your database.
6. Your program calculates all assessment statistics correctly

PHASE 2 (20 points, all carry equal point value).

1. Your program allows the quiz creator/test taker to select options.
2. Your program can edit any question and update whatever contents are stored inside.
3. Your program can delete a question.
4. Your program can move on to the assessment phase at any time.
5. Your program allows the test taker to jump to any question in a correct manner and provides a prompt if they want to edit their answer.
6. Your program allows test takers to advance to any unanswered question, or it presents a prompt menu if all questions are answered and they wish to revise their response for the next question.
7. Your program shows correct answers at the end of the assessment and calculates all statistics correctly.

PART B: (40 points), of which

1. **(15 points)** You have at least 5 testcases in your debug version with each testcase properly commented on what they do.
2. **(15 points)** Your program is free from errors, and you have corner tested your program.

3. **(5 points)** Your program can switch between debug and production version.
4. **(5 points)** Your debug outputs are coherent and logical.

MISCELLANEOUS: (10 points)

1. **(2 points)** Use comments to provide a heading at the top of your code containing your name, Auburn Banner ID, filename, and how to compile your code for both phases. Also describe any help or sources that you used.
2. **(2 points)** Your zipped tarball file should be named like this:

   `project4_ UserID_Firstname_Lastname.tgz`

   For example, `project4_wzc0070_Wan_Cheng.tgz`.

   You will not lose any point if Canvas automatically changes your file name (e.g., `project4_UserID_Firstname_Lastname-1.tgz`) due to your resubmissions.
3. **(3 points)** Your source code is of good quality, easy to read and well-organized.
4. **(3 points)** You must follow the specified user interface/example outputs for each phase.

Note 10: You will automatically get ZERO point if there are compilation errors or warning messages when we compile your source code for both phases. You will lose points if you don't use the specific program file name, or don't have comments, or don't have a comment block on EVERY program you hand in. These will be deducted from your final score, after accounting all other requirements listed above.

II. Programming environment:

Write your program in C++. Compile and run it using AU server (no matter what kind of text editor, IDE or coding environment you use, please make sure your code could run on AU server, the only test bed we accept is the AU server).

III. Deliverables:

You must submit the code of phase 1 and phase 2 in different directory. Put your phase 1 code in directory `~/comp2710/project4_wzc0070_Wan_Cheng/phase1`; put your phase 2 code in `~/comp2710/project4_wzc0070_Wan_Cheng/phase2`.

You must submit a tarred and compressed file – called a tarball, which contains all your .cpp files and header files if you have any. Assuming your project root folder is `~/comp2710/project4_wzc0070_Wan_Cheng`,

use these commands below to create a compressed file for submission:

```
cd ~/comp2710
tar vfzc project4_wzc0070_Wan_Cheng.tgz project4_wzc0070_Wan_Cheng
```

IV. Late submission penalty:

- Late submissions are not accepted and will results in a **ZERO** without valid excuses, in which case you should talk to Dr. Li to explain your situation.
- GTA/Instructor will not accept any late submission caused by internet latency.

V. Rebuttal period:

- You will be given a period of **2 business days** to read and respond to the comments and grades of your homework or project assignments. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

Good luck y'all! And WAR EAGLE!