TCP1201 Objected-Oriented Programming and Data Structures
Assignment 1 Part 1

DUE DATE: 09 Feb 2017, 12:00pm (noon)

A. Outline

This assignment contributes 10% of total subject mark. The assignment consists of only one question. Every student submits one assignment individually. Interview will be conducted.

You are strongly advised to submit the assignment in time even though it is incomplete. For late submission, 2 marks will be deducted for each day of delay.

No extension of dateline will be entertained so that the lecturer would have sufficient time to assess and release the mark before the final exam. Exact interview date and time will be scheduled after the due date.

Make sure your program code can be compiled under GCC 5.3 or CodeBlocks 16.01.

B. Objectives

1. Write a C++ program that utilizes multiple classes.
2. Write a C++ program that utilizes inheritance in a practical manner.
3. Write a C++ program that utilizes polymorphism in a practical manner.

C. Problem Statement

You have been assigned to create a **Quiz Bank** for general knowledge on science and technology. There are three types of quizzes:

1) True/False
2) Multiple Choice (variable number of choices)
3) Short Answer (all answers are a single word)

You will write an abstract class **Question**, that "keeps track of" information common to all three types of questions. All questions will have a String that stores the question as well as a point (score) value. You will then write three classes: **QuestionTF, QuestionMC,** and **QuestionSA**, all of which **inherit from Question**.

You must also create a class called **Player**, that keeps track of the user playing the game. This class should store the first and last name of the player, as well as the number of points the player has. The player initially starts with 0 points.

Initially, your program will ask the user to enter their first and last name, followed by the name of the file storing the questions and answers. The format of this file is given in **Appendix I**. Then, your program should ask the user how many questions they would like for practice (Your program should prompt the user with the maximum allowed, based on how many questions were in the input file they entered.). Ensure your program can handle a situation where the user doesn't enter a valid value. Then, your program should prompt the user with the number of random questions from this database that they requested. **Do NOT ask duplicate questions!**

If the user answers the question correctly, they gain the number points at which the question is valued. If they answer incorrectly, they lose the same number of points.

A sample run of he program is given in **Appendix II**.

D. Requirements

1) Your program must utilize the classes specified above: **Question, QuestionTF, QuestionMC, QuestionSA, and Player.**
2) Your program must error check the response to the number of questions the user wants, checking for both the type of information entered as well as whether or not the value is valid.
3) Your program must ask the proper number of questions randomly from the database.
4) Your program must keep score and prompt the user with how many points they earned or lost after each question, as well as what their total score is at the end of the game.
5) Your program has a reasonable object oriented design. (There are many ways to achieve this, but your program will lose credit even if it works if there is poor object-oriented design.)

E. Bonus points

You may add enhancements to gain some extra bonus points for this assignment. These include (but are not limited to):

1) More error checking than prescribed above. (ie. checking to see if the file entered exists)
2) Allowing users to practice on particular types of questions.
3) Keeping track of past users' performances and storing a list of "best" players.

If you add enhancements, make sure that your program still works on files with the format described in Appendix 1.

Feature Sheet & Evaluation Criteria

Criteria	Item
1. Program Design (5 marks)	1.1. UML class diagrams [0.5m]
	1.2. Style (indentation, self-documentation, identifier) & Modularity (small size functions/methods) [0.5m]
	1.3. Separation of interface and implementation (.hpp and .cpp) [1m]
	1.4. Definition of the classes - choice of attributes/behaviors and the access privileges [1m]
	1.5. Inheritance [1m]
	1.6. Polymorphism [1m]
2. Program execution (5 marks. 0 if unable to compile or run)	2.1. User friendliness (input & output sufficiently self-explain) [1m]
	2.2. Correct program features and output (all attributes must be shown during listing) • Obtain questions from the database (text file) [1m] • Error checking on the response to the number of questions [0.5m] • Display the proper number of questions <u>randomly</u> from the database. [1m] • Keep score and prompt the user with how many points they earned or lost after each question. [1m] • Keep track and display their total score [0.5m]
3. Bonus [3m]	Bonus features stated in **Section E** or other enhancements to the program.
4. Interview (0 mark for the assignment if fail to be present for interview)	3.1. Fluency in using the program
	3.2. Ability to explain code
5. Plagiarism, late submission, or multiple submission	0 mark for the whole assignment

APPENDIX I

Question File Format

The first line of a question database file will contain a single integer n, the number of questions in the file. The following n sets of data contain information about each question.

The first line in each set of data will contain a string that indicates the type of question followed by a positive integer indicating the point value of the question. The string is guaranteed to be one of the three following: "TF", "MC", or "SA".

The second line in each set of data will contain a string that is the question. This string will definitely contain spaces – you must read in the whole line in order to read in the whole question.

If the question is a true/false question, then the third line of the data set will contain either contain the string "true" or the string "false", depending on the answer to the question.

If the question is a short answer question, then the third line of the data set will contain a single string without spaces indicating the answer to the question.

If the question is a multiple choice question, then the third line of the data set will contain a single positive integer, k, between 2 to 6, indicating the number of choices for the question.

The following k lines will contain each of the possible answers. These answers could have spaces in them – you have to read in the whole line. (These lines will correspond to choices A, B, C, etc.)

The last line of a multiple choice question will contain a string storing the correct answer. This string will be a capital letter corresponding to the correct answer choice. (Since there are at most 6 choices, this will always be either "A", "B", "C", "D", "E", or "F".)

Sample Input File (sample.txt)

```
3
TF 5
Pluto is classified as a dwarf planet within our solar system. (true/false)
true
MC 10
Who created Wikipedia on the World Wide Web?
6
Steve Job
Bill Gates
Jimmy Wales
James H. Clark
Tom Davis
Charles Baggage
C
SA 20
The earth is surrounded by a blanket of air. What is it called?
Atmosphere
```

APPENDIX II

Sample Program Run (User's answers in bold and underline.)

What is your first name?
John
What is your last name?
Hopkins
What file stores your questions?
Quiz1.txt
How many questions would you like (out of 3)?
5
Sorry, that is too many.
How many questions would you like (out of 3)?
Two
Sorry, that is not valid.
How many questions would you like (out of 3)?
3

Question 1 (10 points)
Who created Wikipedia on the World Wide Web?
A) Steve Job
B) Bill Gates
C) Jimmy Wales
D) James H. Clark
E) Tom Davis
F) Charles Baggage

Answer: **E**
Incorrect, the answer was Jimmy Wales You lose 10 points.

Question 2 (20 points)
The earth is surrounded by a blanket of air. What is it called?

Answer: **Atmosphere**
Correct! You get 20 points.

Question 3 (5 points)
Pluto is classified as a dwarf planet within our solar system. (true/false)

Answer: **true**
Correct! You get 5 points.

John Hopkins, your game is over!
You final score is 15 points.
Better luck next time!

Assignment 1 Part 2

A. Problem Statement

Modify and expand the scope of assignment 1 Part 1 with the following features:

1. Create a template pointer-based linked list class (LList). Recode assignment 1 to replace at the vector / array used to store the **Question** objects with LList.

   LList must have

a) A default constructor.
b) A destructor.
c) A copy constructor.
d) An isEmpty() function which checks whether the list is empty.
e) A getLength() function which returns the size of the list.
f) An insert() function which inserts a list node to the end of the list.
g) An sortedInsertDesc() function which inserts a list node in the descending order.
h) An erase() function which removes the list node specified by an index number.
i) A retrieve() function which retrieves data (or info) from the list node specified by an index.
j) A randomize() function which randomizes the order of the nodes in the LList

Note:
You can add other private functions only to LList if needed.
The index number of the first node of LList is 1, not 0.

2. In addition to the features stated in Assignment 1, your pogram should have the following additional administrative features to allow the Administrator of the Quiz Bank to perform the following tasks:

a) Add new questions to the database.
b) Remove questions from the database.
c) Search for questions that contain specific word / phrase and edit the particular question.
d) Track the top ten high scores by storing the players' names and scores *sorted by score percentage in descending order*. To fulfill this feature, create a class `HighScore` and use the `LList` to store the list of high scores into the LList by calling `SortedInsertDesc()` method.

   Note:
   For features 2(a), 2(b) and 2(c), the file database should be updated accordingly.

B. Bonus points

Your program can store more than 1 set of Quiz questions (eg. General Knowledge, Science, Mathematics) and allow users to choose which set of quiz that they would like to attempt.

Feature Sheet & Evaluation Criteria

LList (3.5 marks)	
Constructors and destructors	0.5
Public functions: `isEmpty()`, `getLength()`, `insert()`, `erase()`, `retrieve()`	1.5
Public function: `sortedInsertDesc()`	1
Exception handling	0.5
Program execution (6.5 marks. 0 if unable to compile or run)	
All features from Assignment 1 (`Question` *objects stored in LList*)	1.5
Add new questions to the database.	1
Remove questions from the database.	1
Search for questions that contain specific word / phrase and edit the particular question	1.5
Keep and display the top ten high scores in descending order	1
The text file that stores the quiz questions is updated when questions are added, removed or edited.	0.5
Bonus:	
Program can store more than 1 set of Quiz questions and users can choose to attempt a specific set of Quiz question.	2

F. Submission Format

1. The zip should contain all your **UML class diagrams** (in pdf format), your **source code files** (*.hpp and *.cpp). Do not attach any .exe file as some mail servers such as gmail would reject it.