COMP 2710: Project 1 – Phase 2: Design (35 points)

Goals of Project 1 - Phase 2:

- To design a singly linked list.
- To learn the function-oriented design approach.
- To design function prototypes.
- To learn how to design algorithms using pseudo-code.
- To design unit and system testing cases.

1. Overview

From your analysis performed in Phase 1, you can design a class diagram, a system sequence diagram, and system-level test cases in this phase. Please do **NOT** intend to implement the entire program in a single main function.

1.1 Data Structures (6 points)

You must design important data structures. **Note:** You must use **singly linked list** as one of your data structures. You will lose points if you do not use singly linked lists**.**

Note: Suggested structures

You will use structures in this assignment. You will identify which data structures to use in your design. Obvious basic structures will be Menu, System, Character, Scoreboard, Scorefile, Encounter, and Puzzle. You are free to other data structures as you see fit.

1.2. Function Prototypes (8 points)

1.2.1 Requirements

You must design a list of function prototypes for your program to be implemented in the next phase (Phase 3 – Implementation) of this project. Please refer to

http://en.wikipedia.org/wiki/Function_prototype

for details on function prototypes.

Each function prototype must contains the following items:

- Function name
- A list of parameters, including parameter names and their data types
- Return data type
- Purpose of the function

1.2.2 Suggested Functions

A well-done implementation will produce a number of robust functions, many of which may be useful for future programs in this course and beyond.

Remember good design practices include:

- A function should do one thing, and do it well

- Functions should NOT be highly coupled

Some potential functions:

- A few functions to deal with the Menu issues, handling basic user screw-ups (choosing an option out of bounds).
- A few functions for the system, which instantiates the other functions and runs encounters. Receives input from the Menu functions.
- A few function for Encounters, providing the basic framework for a generic encounter (with additional information feed in from system as the encounters occur).
- A Puzzle function which operates with Encounter.
- A HighScores data type which loads, sorts, and collects the high scores.
- A few functions for Characters, keeping track of all the data relating to the character.

1.3 Algorithms (4 points):

You must design important algorithms for the above function prototypes. You may use pseudo code to present your algorithms. Please refer to

http://en.wikipedia.org/wiki/Pseudo_code

for details on pseudo-code.

1.4 Testing Cases (17 points):

1.4.1 (12 points) The System at Large

System-level tests are designed to fully exercise a program as a whole and check that all software components of your integrated system function properly.

For the system at large. In other words, describe inputs for "nominal" usage. You may need several scenarios. Also, suggest scenarios for abnormal usage and show what your program should do (for example, entering a negative number for a menu might ask the user to try again). Please follow the format of the following table below to prepare your test cases at the system level.

Pitfalls:

- No purpose.
- Only offer testing ideas. No expected input and output.
- Expected inputs are not specific.
- Expected outputs are not specific.

Test Cases:			
What	**Input**	**Expected Output**	**Actual Output**
Entering negative number for a menu.	-1	"The selected option does not exist. Please try again."	
Entering a letter for a menu.	G	"The selected option does not exist. Please try again."	
Leaving the menu option blank.		"Please select an option."	
Selecting a menu item that does not exist.	*	"The selected option does not exist. Please try again."	
If user neglects to enter a valid username & password.	Username: aaaaaa Password: 111111	"Error – No user exists. Please enter a valid username and password."	

1.4.2 (5 points) Unit Testing

Prepare a test driver for each function. (Later, these tests can be automated easily using a simple driver function in the function). Remember, test cases are specific, repeatable, and should have a clearly defined result. Testing functions with randomization may require specialized drivers.

2. Deliverables

Please submit your project design through the Canvas system (e-mail submission will **not** be accepted). You need to submit your design document as (1) a C++ source code file named project1-design.cpp and (2) pdf file named project1-design.pdf.

- **project1-design.cpp:** this file contains data structures, function prototypes, and test drivers.
- **project1-design.pdf:** your pdf file should contain the algorithms (i.e., pseudo-code) and the design of system-level tests.

3. Grading Criteria

3.1 (6 points) Data Structures

- (2 points) structures
- (2 points) other data types
- (2 points) singly linked list

3.2 (8 points) Function Prototypes

- (2 points) Function name
- (2 points) A list of parameters, including parameter names and their data types
- (2 points) Return data type
- (2 points) Purpose of the function

3.3 (4 points) Algorithms

3.4 (12 points) System-Level Testing

- (4 points) nominal usages
- (8 points) abnormal usage

3.5 (5 points) Test Drivers

4. Sample Usage

```
        What's your name? Bob


        ============================================================
        |                    Welcome, Bob!                         |
        ============================================================

   1) Start a New Game of Dunstan and Dragons!
   2) View top 10 High Scores
   3) Quit

        Please choose an option: 2

   The top 5 High Scores are:

   Win 1337
   CaseyZZZ 625
   JonnieKill 400
   Bob 75
   Daisy 33
   -no more scores to show-


   1) Start a New Game of Dunstan and Dragons!
   2) View top 10 High Scores
   3) Quit


        Please choose an option: 1

Entering the Dungeon…

You have:

intelligence: 20
time: 25
money: $11.00

You are 20 steps from the goal.  Time left: 25.

         1) Move forward(takes time, could be risky…)
         2) Read technical papers (boost intelligence, takes time)
```

```
        3) Search for loose change (boost money, takes time)
        4) View character
        5) Quit the game


      Please choose an action: 4



You have:

intelligence: 20
time: 25
money: $11.00

You are 20 steps from the goal.  Time left: 25.

        1) Move forward(takes time, could be risky…)
        2) Read technical papers (boost intelligence, takes time)
        3) Search for loose change (boost money, takes time)
        4) View character
        5) Quit the game


      Please choose an action: 2

You read through some technical papers.  You gain 3 intelligence, but
lose 2 units of time.

You are 20 steps from the goal.  Time left: 23.

        1) Move forward (takes time, could be risky…)
        2) Read technical papers (boost intelligence, takes time)
        3) Search for loose change (boost money, takes time)
        4) View character
        5) Quit the game


      Please choose an action: 1

You move forward one step, and…

NOTHING HAPPENS!

You spent one unit of time.

You are 19 steps from the goal.  Time left: 22.

        1) Move forward (takes time, could be risky…)
        2) Read technical papers (boost intelligence, takes time)
        3) Search for loose change (boost money, takes time)
        4) View character
```

```
        5) Quit the game


      Please choose an action: 1

You move forward one step, and…

YOU FIND SOME PAPERS TO GRADE.

You spent two units of time, but gained $3.00!

You are 18 steps from the goal.  Time left: 20.  You can move forward
or backward.

        1) Move forward(takes time, could be risky…)
        2) Read technical papers (boost intelligence, takes time)
        3) Search for loose change (boost money, takes time)
        4) View character
        5) Quit the game


      Please choose an action: 1

You move forward one step, and…

PUZZLE: It's a riddling imp.  I hate riddling imps.  But fine, he asks:
"Find the product of 8 and 8!"

   1) 16
   2) 64
   3) 256
   4) Uh…uh… no?

Choose wisely: 4

The imp cackles "Oh yes.  Yes indeed.  Now you die."

TIME HAS FALLEN TO ZERO.  YOU DIE.

<Print Score, adjust high scores>
```