CS250 Assignment 2
Airline Reservation System

Date assigned: Monday, February 9, 2015
Date due: Wednesday, February 18, 2015
Points: 30

You've been hired by the Computair Airlines Company to work on part of their reservation system. They already have a program that the airline representatives use for advanced reservations. This program generates a file of reservation information about a flight.

Your job is to develop a program that can assign seats to the reservation and help the desk representatives get last minute information about the flight.

The first thing that your program must do is to assign a seat to each reservation. The easiest way to keep track of information about the flight is to store the following information about each seat. You must use a struct.

- The last name of the person assigned to the seat.
- The first name of the person assigned to the seat.
- Whether or not the seat is available.

Each of the aircraft that the airline flies has 30 rows of 6 seats. The rows are numbered from 1 to 30 and the seats are labeled from A-F. Seats A and F are window seats and seats C and D are aisle seats. So, you would want to store a two dimensional array of information about the seats.

You have an input file with the following information about the reservations:

- First name
- Last name

You are to assign seats to the reservations in the order that they are read from the file with the following stipulation; you are to assign all of the window and aisle seats first before assigning the middle seats. So, for the following input:

```
Sally Smith
Jim Smith
Tim Smith
Molly Smith
```

Sally Smith would be assigned seat 1A, Jim would be assigned seat 1C, Tim would be assigned seat 1D, Molly would be assigned seat 1F and John would be assigned seat 2A. So, you would not assign seats 1B and 1E (the middles) until you have filled all of the aisle and window seats on the flight.

Now, you need to provide a menu of functions that the workers at the airport can use to generate information about the flight. This menu should give the user the following options:

1. Output the number of free seats on the plane
2. Output the empty seat closest to the front of the plane
3. Output the first and last name of the person in a given seat
4. Output a diagram of the seat assignments to the screen
5. Quit

The program should continue to allow the user to enter menu choices and receive information until the user chooses to quit. You are given the freedom to present the information in anyway that you think is appropriate, except for option 4, which must look like the option below.

Note: You must use the airlines seating number conventions when you input and output seats. For example, you must use seat 1A not seat 0 0 (even though your program will store that seat as 0 0 internally).

Input: The file will contain lines of first and last names with a space between them and a new line between each passenger. For example:

```
Jeff  Weaver
David Wells
Michel Hernandez
Jorge Posada
Chris Widger
Erick Almonte
Jason Giambi
```

Your **output** is to look exactly like the following:

```
**************************
Computair Airlines Company
**************************

Passengers have been assigned seats.

Select an option from the following menu:

1. Output the number of free seats on the plane
2. Output the empty seat closest to the front of the plane
3. Output the first and last name of the person in a given seat
4. Output a diagram of the seat assignments to the screen
5. Quit

Option: 4

     A           B           C           D           E           F

1    Weaver                  Wells       Hernandez               Posada
2    Widger                  Almonte     Giambi
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

(all the way to 30. Then you must display the menu again)
```

Goals for Assignment 2

1. Implement a C++ program that uses files, 2D arrays, and structs.
2. Practice modular programming by using well defined functions
3. Break up a program into well-defined functions. This is the first assignment where you choose the functions. It is important that your functions be small, and focused on a particular task.

Notes:

1. Your main function is to be mainly variable declarations and function calls.
2. Test your program one function at a time.

To complete this assignment you must submit the following:

1. An electronic copy of your program on Grace

a) Create a new C++ solution in Visual Studio 2010. Your solution must be called **PUNetID-Assignments**. For example, mine would be called khoj0332-Assignments. This solution will hold all of your assignment projects for CS 250. You are starting over for CS250, meaning do not include any projects from CS150.

b) Your project for this assignment must be named **02_AirlineReservation**. It is vital that you name your solution and your project correctly!

c) Type your program (fully documented/commented) into the project. You need to follow the coding standards from the CS250 Web page. These coding standards have been modified to include additional C++ language features introduced in CS250, so please be sure to read the new coding standards.

d) Pay attention to the example output. Your program's output must look **exactly** like the sample output. The spacing and newlines in your output must match exactly.

e) Make sure that your program builds without errors & warnings and runs correctly. If you get any errors or warnings, double check that you typed everything correctly. Be aware that C++ is case-sensitive. You will lose 10% if there are any warnings and 70% if your program does not build successfully.

f) Once you are sure that the program works, it is time to submit your program. You do this by logging on to Grace and placing your complete solution folder in the **CS250-01 Drop** folder.

g) The solution must be in the drop folder by the time class starts on the day the assignment is due. Anything submitted after that will be considered late.

2. A hard copy of your program

a) The hard copy must be placed on the instructor's desk by the time class starts on the day that it is due.

b) The hard copy must be printed in color, double-sided, and stapled in the upper left corner if your solution contains multiple pages.

c) Your tab size must be set to 2 and you must not go past column 80 in your output.

Remember, if you have any problems, come to me straight away with your project on a flash drive or on Grace. Good Luck!!!! ☺