**Assignment #2**

*Problem Solving and Programming in C++*

***Department of Computer Science Old Dominion University***

**Objectives**: This assignment will give you an opportunity to explore the process of dividing a **C++** program into modules and using the project support components of a C++ IDE to manage your modules.

**General Instructions**: Read the problem description below and reorganize this program in C++. The files for this assignment are provided under the folder for this assignment. You will be submitting two separate projects. See Part A and Part B for details.

* All of the functions making up this program are complete and in working order except for functions marked with “/// FILL THIS FUNCTION”. You will need to implement these functions.
* The major challenge in this assignment is to divide the program into separately compiled modules.
  + A candidates module with files ‘candidates.h’ and ‘candidates.cpp’, containing code dealing specifically with manipulating the reservations array.
  + A openings module, consisting of files ‘openings.h’ and ‘openings.cpp’ containing code dealing specifically with the opening position records.
  + An optional general functions module with the files ‘generalFunctions.h’ and ‘generalFunctions.cpp’ containing code that is not specifically geared towards candidates or openings.
  + Consult the comments in the provided code for additional details.
  + There are some functions that are not specifically related to any of these modules, but are called by or contain code needed by certain modules You should apportion these functions to the modules in a way consistent with the goals of high cohesion.

# Problem description:

# A new startup company has been left without a database after a falling out with the previous database administrator. As an intern, you have decided to impress management by creating a program that uses some of those files to aid company recruiters in managing openings and potential candidates. The program must have the following options: Add Opening, Delete Opening, Show Openings, Match Candidates with all Openings, Add Candidate, Show Candidates, Show Candidates by Position, Filter Candidates By Rating, and Clear Screen.

# 

# Input

Input to the program is taken from several text files containing data regarding job openings and registered candidates. These are included, but your program should function even if the information in the data files were to change with the format being the same. The first line in every file is the number of data points (lines) in the file.

**Output**

The output should be regular updates to the 2 files, as well as interactive menus display the file contents.

**Part A:**

Create a project containing the file ‘main.cpp’. Implement the empty functions marked with “///Fill This Function” comments. You will need understand the purpose of these function in order to implement them. Do this by looking for where the functions are being called and how the function is being used in the program. Also, there may be other function that do similar, though not the same task. These functions may give you a clue how to implement the empty functions.

The program should compile and have expected Input and Output as outlined above. You will not need to split the functions into modules for part A.

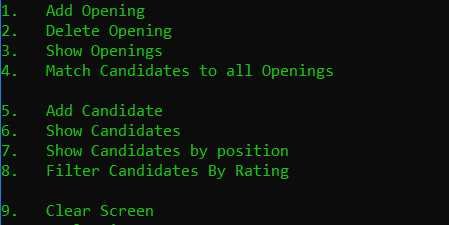
**Part B:**

Create a project containing the files ‘main.cpp’, ‘candidates.cpp’, ‘candidates.h’, ‘openings.cpp’, ‘openings.h’, ‘generalFunctions.cpp’, ‘generalFunctions.h’. Split the functions from Part A into the appropriate modules as outlined in the general instructions. The program should compile and have the same input and output as Part A. In order to determine where a function belongs, look to see what functions it calls, and which functions it is called by. Functions which rely heavily on each other are good candidates for a module. Functions which are associated in purpose may also be candidates for a particular module.

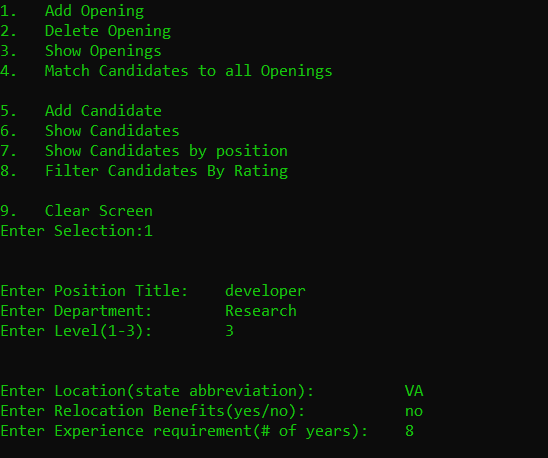
# Submission notes:

* Submit all files from your project **folder**, including the .cpp, .cbp, .h, and input files.
* Be sure that your project compiles on CS Department computers before submission.
* Zip each **folder** and name it as “Assg2A\_cslogin.zip” and “Assg2B\_cslogin.zip”, where the cslogin is your login ID for the computers at the Department of Computer Science at ODU. Bin and Debug folders do not need to be included.
* Submit the zipped file in the respective Blackboard link.

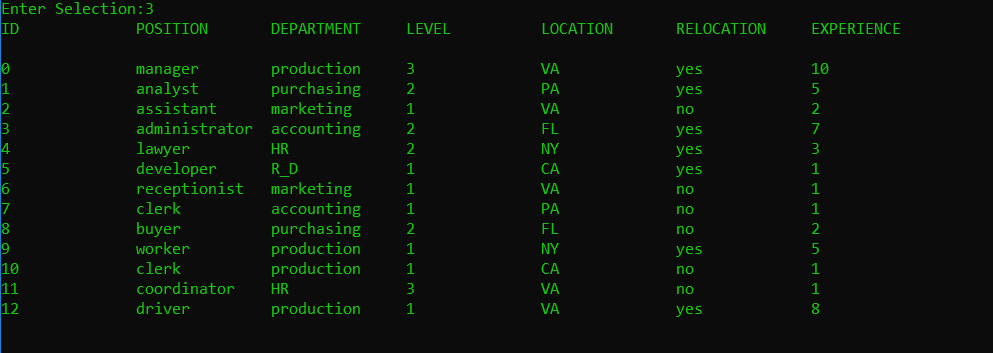
Main Menu



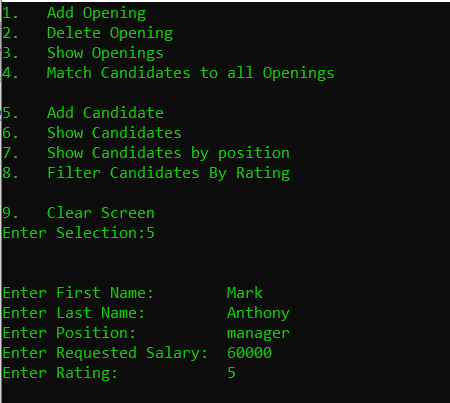
Add Opening



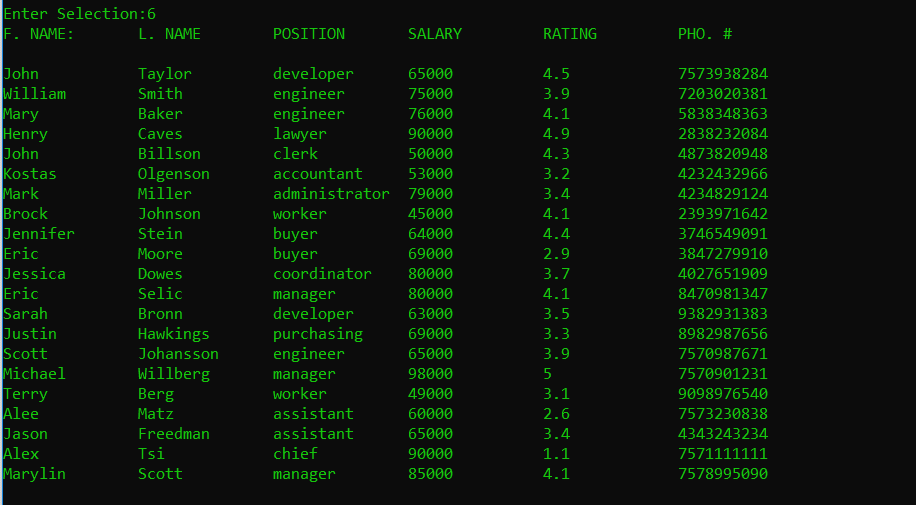
Show Openings



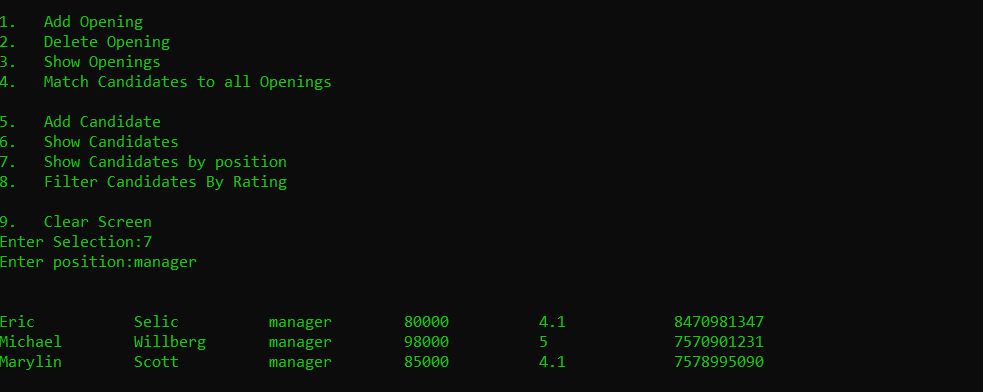
Add Candidate



Show Candidates



Show Candidates by Position



Show By Rating

