

Description:**Due before 11:59pm on Friday, April 22**

In this assignment, you will be developing an administrative interface for our semester game project, by creating and maintaining a linked list (LL) of objects. You must use a linked list to solve this project, in order to receive any credit.

All work must be submitted before the due date.

Hand in: Source Code. Remember to **include the necessary documentation in the source code.**

There is a link under the Assignments tab on Blackboard, for you to submit your work.

Name your documents correctly: Your Source Code and documents should include your first initial and last name. For example: Stewie Griffin would name his source code file – **SGriffin_Project5.cpp**

Write a program that processes an input file containing information on our semester game players. The program creates a dynamic list of players, and provides management **with a menu** with some tools that include:

- Create the list either Forwards or Backwards
- Print the list of all players game history.
- Print the top five player scores (along with alias names, and dates)
- Calculate game statistics:
 - high score (and alias name)
 - low score (and alias name)
 - average score of all games played
 - average difficulty level of all games played.
- Locate alias of player (similar to locateID function in lab 11)
- Remove a player history from the list, based on a player alias.
- Add a game history item to the list.

All changes should be written to the gameHistory.txt file before exiting.

The input file is the same gamehistory.txt input file from project 4. Fortunately, most of the coding design is similar to lab 11 and the sample code detailed in class over the last two weeks.

Make sure to include your name and lab CRN in the appropriate places. To receive full credit, you must **write functions for each of the tasks, and your program MUST perform all the required functionality. If it doesn't run, you will not receive any credit.**

POINT BREAKDOWN

- Menu (10)
- Create forwards (5)
- Create backwards (5)
- Print players. (5)
- Print top five players (5)
- Game statistics: (20)
 - high score (and alias name)
 - low score (and alias name)
 - average score of all games played
 - average difficulty level of all games played.
- Locate alias of player in list (10)
- Remove player based on a player alias. (20)
- Add player history item to list. (20)

All changes should be written to the gameHistory.txt file before exiting.