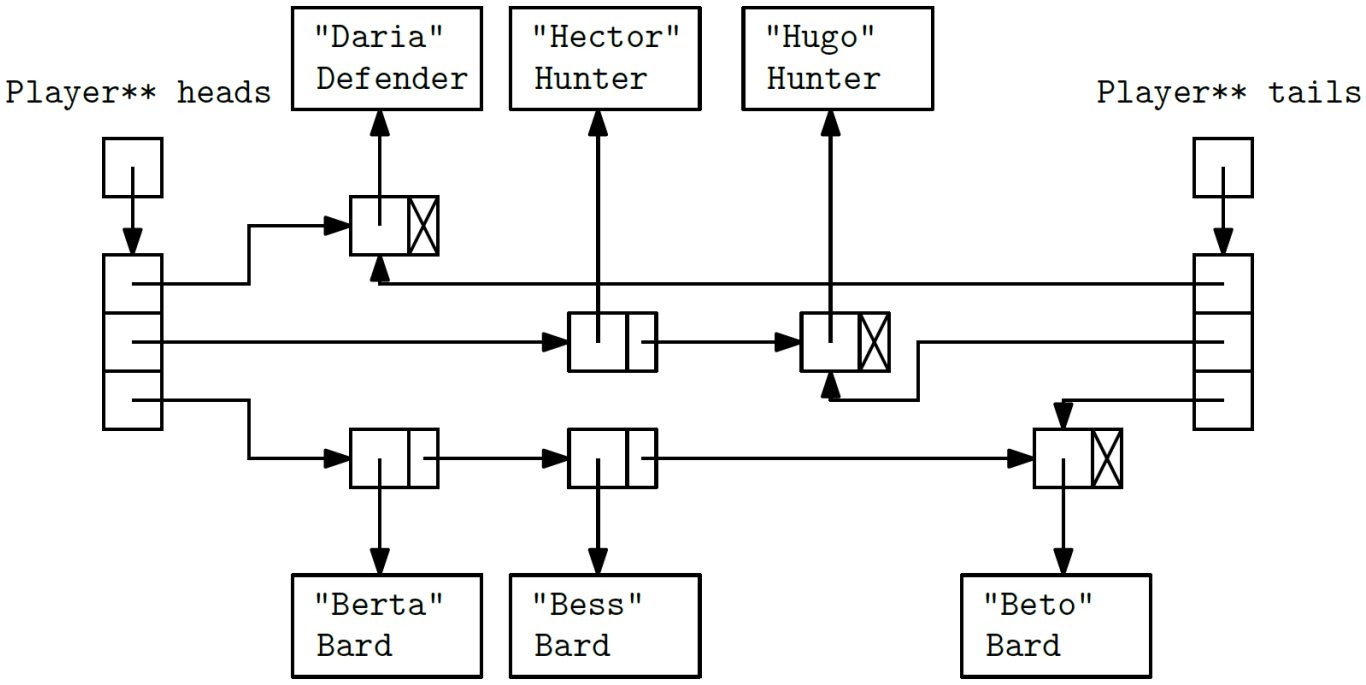
CSCI 2380 **Spring 2025** **Mr. Gustavo Dietrich**

Homework 7: Queues using linked-lists **Due Date:** 3/27/25

In this second iteration of the "looking for group" matchmaking queue, you'll eliminate the potential slowness caused by searching deep into the array for the front most player of each role. To do so, you'll use three linked-list-based queues, one for each player role.



The following files have been given to you:

1. A C++ header file (lfgqueue.h) declaring the **LFGQueue** class.

2. A C++ header file (player.h) declaring the **Player** class.

3. A C++ header file (player.cpp) implementing the **Player** class.

4. A C++ source file (main.cpp) containing a main() function with tests.

Create new C++ source file named **lfgqueue.cpp** that implements the classes declared in lfgqueue.h so that lfgqueue.cpp and the provided files compile into a program that runs with no failed tests.

Submit just the source code of **lfgqueue.cpp**. You don't need to submit the main.cpp nor the header files because I will use my own lfgqueue.h, player.h, player.cpp, and main.cpp files to evaluate your lfgqueue.cpp file.

Review the examples discussed in class and the textbook to get an idea of what you need to do. Analyze carefully the tests because that will help you understand how the methods that you need to create work.

Do not hesitate to use the corresponding topic in Discussions to post your questions/doubts about this assignment. I will reply as soon as I can.

IMPORTANT:

Make sure your program compiles and executes in full (it should pass all the tests included in main()).
You must submit ONLY ONE solution per team.
Your program must be well commented, use meaningful identifiers, and use indentation to improve its readability.

Your program must have the following comments at the top:

```
//*************************************************************************
// Team #              CSCI 2380              Spring 2025              Homework # 7
// First and Last Name
// First and Last Name
//
//*************************************************************************
```

When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.
Paste the link to your final solution along with your source code in the textbox opened when you click on Create Submission before you click on Submit.

The following is the basic criteria to be used to grade your submission:

You start with 100 points and then lose points as you don't do something that is required.

-14 : Incorrect implementation of LFGQueue()
-14 : Incorrect implementation of string push_player(Player* p)
-14 : Incorrect implementation of front_player(Player::Role r)
-14 : Incorrect implementation of pop_player(Player::Role r)
-14 : Incorrect implementation of front_group(Player** group)
-14 : Incorrect implementation of pop_group()
-14 : Incorrect implementation of void size()
-10 : hard-coded for each player (very inefficient) when you should use the role to access the corresponding player
-20: Program crashes when executed
-5: Unnecessary statements in your code
-10: Missing/too few comments
-40: Program does not compile
-20: Incorrect/missing source code
-20: Incorrect/missing link to your Repl.it solution
-100: No team contribution
-100: No Did not submit a join link
-100: The code submitted is not your creation (you got it from a web site or another person)
-10: Late
-100: No submission.
-101: Incorrect submission. Talk to me please.