Suggested steps for implementing PA#5

1. Review the Number class demo, especially the implementation of the member class functions. Make sure you can compile and run it. (Canvas > Files > NumberList demo).

2. Download the provided files from zybooks: Driver.cpp, MovieList.h, and MovieNode.h

3. Review **MovieNode.h**. Don't change it. It uses some different syntax/style:
   - It doesn't have "using namespace std", so it must put std:: in front of string.
   - It provides a constructor for MovieNode. This means when you declare a MovieNode struct, you must do it this way:
     `MovieNode node("Star Wars",nullptr);` or
     `ptr = new MovieNode("Jaws",nullptr);`
     (you can put any appropriately typed pointer in the second position).

4. Review **MovieList.h**. It's similar to NumberList.h, except the MovieNode is defined in a separate file (MovieNode.h). It contains the functions you need to define.
   - What is ostream& out? It's a parameter that allows an output stream like cout or fout to be passed in as an argument. Use it like cout in your display functions to do your output (don't use cout!!).

5. Create the **MovieList.cpp** file. You will define the member function definitions in there. (Be sure to include MovieList.h).
   - Note if you include <iostream> using namespace std, then you do NOT need to put std:: in front of the library-defined names (but you can also leave them in if you want).

6. Review **Driver.cpp**. It's ok if you don't understand the code exactly. It contains the unit tests from zyBooks. When you compile and run it with your MovieList.cpp, it will tell you if you passed or failed each test, and what was expected. You do not need to compare your output to a sample output. If you prefer the old style driver that calls the functions and outputs to the screen, and you compare it to expected output in a provided text file, I have linked to OldStyleDriver.cpp and sample_output.txt in the Canvas assignment.

7. Implement the MovieList functions in this order (recommended). You may re-use code from the NumberList demo provided on Canvas.
   - constructor
   - destructor
   - addToTop
   - display (now you can start compiling and testing)
   - addToBottom
   - nextLarger
   - displaySorted

   I'd wait until after Wednesday's class to do these:
   - remove (this changes the list, we'll cover this operation Wed 10/28)
   - moveToTop (this changes the list, you can call remove)
   - moveToPosition (this changes the list, you can call remove)

8. You may need the `-std=c++11` flag in order to get it to compile.
   `g++ -std=c++11 Driver.cpp MovieList.cpp`

9. You can start submitting your MovieList.cpp file to zybooks as soon as you pass at least one of the test cases in Driver.cpp (no makefile this time).

Please compile your code in a Linux/Unix environment.