**Lab 3 – Collections using Object Composition**

This lab will give you more experience with creating and using classes, and add in concepts such as object composition, arrays of Objects, and using separate header and source files for class definitions.

**Your Program**

The Movie class created for Lab 2 offers a convenient place to start in this one. We’re going to want to start thinking about organizing our code a bit more though, so start by separating the Movie class definition into a separate header file.

Once the Movie class is created and moved into its own file(s), I recommend testing it thoroughly to make sure all of the parts are working before moving on. You will need to remember to include the header file in any other file where you intend to make reference to the class.

After the Movie class is created, you will create a second new class, MovieDatabase, this one with its own header AND source files (MovieDatabase.h and MovieDatabase.cpp). The MovieDatabase class will have (at least) the following private attributes:

* An array, Movies, capable of storing 100 Movie objects.
  + This is inefficient, and limited, but we’ll fix it later in the semester.
* An integer, movieCount, that counts how many movies have been added into our database.
  + While 100 is the MAX size of the array, the ACTUAL size of the filled part of the array is important to keep track of, and this variable serves that purpose.

Besides any accessors and mutators, your MovieDatabase should include methods designed to accomplish the following tasks:

* Add a movie to the database
  + This function will need to allow the user to enter all the required information to set the values of a movie in the array. You will need to know how many movies are already in the array to determine what index of the array you should edit. For instance, if there are 10 movies already, then the new movie would go to the 11th position, or Movies[10].
* Display the current number of movies in the database.
* Print out the titles of all of the movies in the database.

In the main, you should declare an instance of your database class, and implement a menu driven program that allows the user to choose to add movies, display the number of movies, or list the titles (by calling the associated methods). Test your program thoroughly, and make sure that it works reliably.

When you are done, submit the completed .cpp and .h files through the Blackboard submission tool. Please name your main source file [FirstInitial][LastName]\_Lab3.cpp, substituting your own name. For instance, my submission would be LSilcox\_Lab3.cpp.

**Challenge** [25 pts Extra Credit]

Implement another method in the MovieDatabase class that accepts a movie title as a string argument, and searches the array of movies for a match. You can use a technique called a Linear Search for this – start at the first element in the array, and compare the title of that movie to the desired title. If you find a match, call the display function for that movie to print out the data, and then exit the search. If you don’t find a match, iterate to the next movie in the array and try again, until you either find a match or run out of movies to check (it’s not efficient, but it works). If no match is found, you should display an error message indicating such.

In the main, you should update your menu to allow the user to use your new search method.