**Program 1B**

**Due**: October 3 (by class)

**Worth**: 40 pts.

**Purpose**: This program explores the creation of a simple class hierarchy including (limited) use of polymorphism and LINQ to produce simple reports.

In this assignment, you will use the Library class hierarchy developed for Program 1 (either your solution or your instructor's) in a simple test program. Instead of using partially-filled arrays of library items and patrons, your program will use the **List**collection described in Chapter 9. You must create a **List** of**LibraryItem** objects filled with a least two instances of every concrete class from the hierarchy. You will need to add at least two **LibraryMagazine** objects with the same title (but different volume/number). You must also create a **List** of **LibraryPatron** objects with a least 5 patrons.

The detailed **public** requirements for the classes in this assignment appear below.

* Display the list of items to the console immediately after construction (none should be checked out yet) and then pause and wait for the user to enter a key.
* Check out at least 5 items and display the list of items again. Pause and wait for the user to enter a key.
* Using LINQ, select all the items from the list that are checked out and display the resulting items and the count of checked out items (hint: use the Count method on the LINQ result variable). Pause and wait for the user to enter a key.
* Using LINQ, filter the previous result set to select only checked out **LibraryMediaItem**s. Display the results to the console and then pause and wait for the user to enter a key.
* Using LINQ, select the *unique* titles of the **LibraryMagazine** objects from the list and display the titles to console. Pause and wait for the user to enter a key.
* For each item in the list, calculate what the late fee would be if it were returned 14 days late. Display the item's title, call number, and the late fee to the console. Pause and wait for the user to enter a key.
* Return all the checked out items. Show that the count of checked out books is now zero using the earlier LINQ result variable. This will demonstrate that LINQ uses *deferred execution*. Pause and wait for the user to enter a key.
* For each **LibraryBook** in the list, display its current loan period and then modify it to add 7 more days. Pause and wait for the user to enter a key.
* Finally, display the entire list of items to the console once more.

This assignment will only focus on the test program and not the hierarchy classes. If your hierarchy classes don't work, you will need to use your instructor's solution to Program 1A as a starting point.

Please review the documentation requirements specified in the syllabus that are expected for each file. These include precondition and postcondition comments for each method, property, and constructor. Also, each variable needs a comment specifying its purpose.

Rather than giving me floppy or Zip disks, you will upload **all your files** to using the *Assignments* tool. Using a tool like *PKZIP*, zip your entire Visual Studio project directory into a compressed file and upload the .zip file. Before uploading, make sure you can unzip the file, reopen the project, and execute the application. Remember, all assignments are due at the start of class and no late assignments will generally be accepted. For this assignment, I will grade all files electronically. No printouts are required.

Remember, this is an **individual** assignment. Please be mindful of the syllabus' statement on academic dishonesty. If you are unsure about what constitutes academic dishonesty, **ASK!**