CSCI 145 Intermediate Programming
Semester Project Part C

For this part, you will use the classes you created in part B and move them to modules. (if you had errors, be sure to correct them). You must have one program file containing the working code of your program. **All your classes should be turned into modules.**

For this part of your program, you add the **use of linked lists as running storage for your storage system**. From your part B implementation, you created inheritance, aggregation, and dependency. Many of you also used Lists to implement running storage in acquiring multiple pieces of information. Now you will need to add the use of Linked Lists to your program. You will also need to maintain order in your system…whether alphabetical or numerical. **You must use one of the sort algorithms created and described in class.**

Note: take great care when implementing these types of lists, it is very easy to lose track of your various nodes.

As for more permanent storage, you will use files. The difference between the files used in CSCI 135 and this course, is the use of objects. **You will store and read objects from permanent storage.** You do not have to store every piece of data (although this will be a true test of what you know), you may choose one type of data OR if you have one large object, you are storing all of your data. For instance, if you have media items made up of books and films, you may want to store your list of books to a file. You will need to show storage and retrieval. Remember, you cannot read objects stored in a file…they look like wing-dings (garbage). Choose data that is relevant to your program you should be able to restart the program and data was saved from the previous executions of your program.

Program file: You will maintain your menus from part A and B. Remember to name this class file in a way that represents the actual system you are building (not just SemesterPartC but the name of your program).

This must be a complete, polished, and well organized program that does not crash. If your user instructions are hard to understand or your program crashes grading will stop and you will only receive points gained from that part of the program.

Timing Tip: This project is due the Monday after our final week 14. Have this project complete before the last class of week 14. In this class Moodle will randomly select teams of two where you will switch projects and test them. You will not tell each other what your project is or how it works. You will only watch as they run your project. This will give you the opportunity to see how someone who has never seen your program will attempt to use it. This will give you valuable information to be successful in Part C.

Upload all files using the link provided in Moodle. There is currently a maximum of 10 files.