**CSE 274 – Spring, 2023**

**Homework #2 – Due Sunday, February 26, 2023 by 11:59 p.m.**

**No late assignments accepted for any reason**

**LinkedLists, Sorting, and Generics**

**General requirements (not following any of these could result in a score of 0):**

* **Code is expected to compile and meet basic coding standards for readability**
* **You may use as many Java classes as you need- putting everything in main is NOT a good coding solution. Main method should be in a class by itself.**
* **Use the coding standards linked in your Canvas class under Useful Links. Failure to do so will result in deductions.**
* **Make sure you cite any code not written by you and add your name to the author list if you made changes. Put your name on any file you start from scratch. It is expected that no more than 25% of the code is cited or copied from examples given or websites**

# The Problem

Create and use a Generic Linked List. Create an instance of your LinkedList that stores Media objects. Using the media.txt from Homework 1- read in the Media objects and store them in your LinkedList. Print the results of the list toString() method after completing the input from the file.

Then, call your sort method and print the toString() of the newly sorted list.

* Creating a Generic LinkedList class – you may use either Singly or Doubly linked list- your choice. Note: singly linked list is easier.
  + LinkedList must implement the given List.java interface. You may use the LinkedList from Lab2 as a starting point if you wish or create a new implementation.
  + You may add any additional methods you need beyond those required by the List interface.
  + Implement sort method that will sort the LinkedList provided that the object being sorted implements Comparable<E> interface. Merge sort is suggested as it works well with Linked Lists.
    - Merge sort sources (use as a guide for your merge sort method- do not just cut/paste from these sites)
      * <https://www.educative.io/answers/how-to-sort-a-linked-list-using-merge-sort>
      * <https://www.geeksforgeeks.org/merge-sort-for-linked-list/>
* Use the Media.java, Book.java and Video.java classes from Homework 2. You will need to modify Media.java so it implements the Comparable interface. The additional compareTo() method should return less than, equal to, or greater than based on the unique ID of the Media object.
  + Examples of implementing Comparable
    - <https://www.geeksforgeeks.org/comparable-interface-in-java-with-examples/>
    - <https://howtodoinjava.com/java/collections/java-comparable-interface/>
    - Chapter 9 in Big Java by Cay Horstmann available in oreilly.com
* You can use the code you wrote in homework 1 to read the file, but DO NOT USE STATIC VARIABLES. You can use static methods in the class containing main, but unless it is a constant (final) that class should not contain static variables.

# Expected Output (Bold is for emphasis in this document- you do NOT have to bold any output). For the original list, it may also be in reverse order to that presented. Individual items may also have slightly different presentation, but must include all information about each item.:

**The original list:**

Book [author=Some author, publisher=Some publisher, title=The Last Item, uniqueID=999, type=book]

Video [director=James Cameron, format=Blu-ray, title=Avatar, uniqueID=006, type=video]

Book [author=Mark Twain, publisher=publisher2, title=Tom Sawyer, uniqueID=004, type=book]

Book [author=Margaret Mitchell, publisher=publisher1, title=Gone with the Wind, uniqueID=003, type=book]

Video [director=Baz Luhrmann, format=digital, title=Elvis, uniqueID=005, type=video]

Book [author=Eric Carle, publisher=Philomel, title=The Very Hungry Caterpillar, uniqueID=007, type=book]

Book [author=author, publisher=publisher, title=title, uniqueID=444, type=book]

**The sorted list:**

Book [author=Margaret Mitchell, publisher=publisher1, title=Gone with the Wind, uniqueID=003, type=book]

Book [author=Mark Twain, publisher=publisher2, title=Tom Sawyer, uniqueID=004, type=book]

Video [director=Baz Luhrmann, format=digital, title=Elvis, uniqueID=005, type=video]

Video [director=James Cameron, format=Blu-ray, title=Avatar, uniqueID=006, type=video]

Book [author=Eric Carle, publisher=Philomel, title=The Very Hungry Caterpillar, uniqueID=007, type=book]

Book [author=author, publisher=publisher, title=title, uniqueID=444, type=book]

Book [author=Some author, publisher=Some publisher, title=The Last Item, uniqueID=999, type=book]

Extra credit (up to 15 points): Implement a doubly linked linked list. Provide an additional method to printBackwards(). Make sure the sort method ensures that the list is still correct when printed backwards after sorting. Demonstrate this in your main to get all credits. Note: Most linked list sort algorithms do NOT handle doubly linked lists – only singly linked.

## Grading Criteria

|  |  |
| --- | --- |
| Criteria | Points |
| Expected Output received (sort works and output neat/readable) | 35 |
| LinkedList Generic | 15 |
| LinkedList has all requested methods correctly implemented | 30 |
| File Reading Correctly done with exception handling (try/catch blocks) | 10 |
| Code follows coding standards | 10 |