Langara College

# Department of Computing Science & Information Systems

# CPSC 1150 – Program Design

###### **Lab11: File\_IO**

**Objectives:**

* Write several methods that use and manipulate arrays
* To become familiar with writing to and from files
* To practice method decomposition, arrays, and file i/o

**Instructions:**

1. Create a folder named **Lab11** to store all the files from this lab
2. Create an external documentation file (filename: **Lab11Ext.docx**) to store the summary, algorithm(s), and sample input and output for each problem.
3. All your programs must have good internal and external documentations

**Problems [35 marks]**

**Preamble**

The file releasedates.txt contains a list of video games and their release dates. Each line of the file contains the release date, a tab character, and then the name. The list is currently totally unsorted. The object of today’s lab is to write a series of methods that allow us to perform the following tasks: read contents from a file and store them in array, sort the contents of the array by name or by date, and store the sorted array results back into a new text file. The first few lines of releasedates.txt look as follows:  
04/26/16 Sega 3D Classics Collection

07/14/16 Batman: Arkham Underworld

06/24/16 Tokyo Mirage Sessions ♯FE

**File I/O [10 marks]**

1. [5] Write the method String[] input(String filename) that accepts a file name as a parameter. This method should read in the data from the file and store it into an array, then return this array.
2. [5] Write the method void output(String filename, String[] info) that accepts a String called filename as a parameter. This method should create a new file called filename and write the contents of the array info neatly into the new file.

**Sorting [15 marks]**

1. Write the method String[] sortByMonth(String[] info). This method should sort the array from January release date to December release date and return it.
2. Write the method String[] sortByName(String[] info). This method should sort the array by game name from A to Z and return it.

* **Note: you may not use any of Java’s built in sort methods. Implement one of your favorites that we’ve covered in class so far. Using a built in sorting method will result in a mark of 0 for this portion of the lab.**
* *Hint:**you may use* ***compareTo()*** *method from the string class to compare two strings.*
* *Hint: After you declare the Scanner object:* ***Scanner sc = new Scanner(file)****, you may use* ***sc.useDelimiter(“\n”)*** *indicating that the new line character “\n” will be used as your delimiter for* ***sc.next()****. Recall the default delimiter is the space character, so now* ***next()*** *will stop reading when it sees a new line character instead of a space character.*
* *Hint:**do not re-implement your sorting algorithm in each method, write one sorting method and then just call it in each of the above methods.*
* *Hint: you may find it easier to take the array of information and separate it into two new arrays, one for the list of games and one for the list of dates, and then perform the sort. It is important to make sure that the indexes of the arrays do not get out of sync, or you may lose the correct release dates for the different games.*

**Main Program [5 marks]**

Your main program should call your input method on releasedates.txt.

Your main program should call the sortBydate method, then write the result of that method call into a new file called SortedByDate.txt

Your main program should call the sortByName method, then write the result of that method call into a new file called SortedByName.txt.

**Sample Output**

The first few lines of SortedByDate.txt should look **exactly** as follows:  
01/04/16 Pony Island  
01/05/16 Amplitude  
01/05/16 Hardware: Rivals

The first few lines of SortedByName.txt should look **exactly** as follows:  
10 Second Ninja X

1979 Revolution: Black Friday

7 Days to Die

7th Dragon III Code: VFD[B]

A Boy and His Blob

A.W.: Phoenix Festa

**What to hand in**

* Make sure that your code is **neat**! Each method must have proper internal Java Documentation. **[5 marks]**
* Submit to D2L a zip/archive file containing the .JAVA files you have written for this lab (NOT .CLASS files)
* If you do not zip your file or if you submit the .class files, you will receive **ZERO** for this lab (<http://www.wikihow.com/Zip-Files-Together>)

**When to hand in**

By 10:29 am, Monday, April 5, 2021