CS 112 Programing II

Lab Session 7: File I/O and Exception Handling

Objectives

In this lab session you will be asked to create the class MergeFile which is very similar to the ReplaceText class that was discussed during your previous lectures on *Exception Handling and File I/O*. Upon finishing this lab session, you should be able to:

- Create files using the class File.
- Read data from a file using the Scanner class.
- Write data to a file using the PrintWriter class.
- Create text files from Netbeans and set your program arguments.

Exercise 1

Design and implement a class named MergeFiles which does the following:

i. Reads names stored in files file1 and file2

ii. Merges the two lists of names into a single list

iii. Sorts that single list

iv. Ignores repetitions

v. Writes the sorted, free-of-repetitions list to a new file named mergedFile.txt

vi. The names in all three files are stored as one name per line

vii. The names of the two files containing the names as well as the name of the file to contain the sorted file names are passed as command-line arguments to MergeFiles as follows:

```
java MergeFiles file1 file2 mergedFile
```

Below is the program logic which will guide you through the process of writing your program:

a. Check if the user uses the *correct number of arguments* (show an error message and exit the program if the number of arguments is incorrect).

b. Create an ArrayList object, names

c. Open the 1st file (args[0]) for reading using File and Scanner (show an error message if the file does not exist).

d. Read the contents of the 1st file into the names object

e. Open the 2nd file (args[1]) for reading using File and Scanner (show an error message if the file does not exist).

f. Read the contents of the 2nd file and add them to names

g. Sort names by first converting it to an array of String and then applying the built-in Arrays.sort() method

h. Print the contents of the sorted array to the merged file (args[2]) using File and PrintWriter (show an error message if the file exists already)

i. Skip any duplicate names while printing

j. Close the input and output files (why?)

k. Invoking the constructors new **Scanner(file)** and new **PrintWriter(file)** may throw an I/O exception which must be caught or declared in the calling method i.e., the main. For simplicity, declare throws Exception in your main header.

l. The MergedFiles class can be invoked as follows:

```
java MergeFiles file1.txt file2.txt mergedFile.txt
```

In order to test your program do the following:

1. Create file1.txt and file2.txt in the MergeFiles NetBeans project directory. Put at least five names in each of these files.

1. In NetBeans, go to *Run* > *Set Project Configuration* > *Customize…* > *Arguments*. In the field in front of *Arguments* type the following:

```
file1.txt file2.txt mergedFile.txt
```

2. Now if you run `MergeFiles` from within NetBeans, it will run with those three command line arguments.

2. Run the program, and examine the resultant mergedFile.txt to ensure that the program is working properly.

Exercise 2

3. Delete `throws Exception` from `main()` and try to run the program. You will receive an error. Is this error due an *unchecked* exception or *checked* exception?

4. Rewrite your code using *try-with-resources* which automatically closes the files after use.