ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 1

Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of all students in your group.
2. Each student in the group is required to submit the assignment on moodle.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. **Export your project as follows:**
   a. From eclipse, choose *"Export…"* from the File menu.
   b. From the Export window, choose *General* then *File System*. Click *Next*.
   c. Make sure that your project for this assignment is selected. Make sure that all of its subfolders are also selected.
   d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
   e. When exporting make sure you select *Create directory structure for files*.
   f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
   a. All the group members should submit the same zip file.
   b. The file name is very important and should follow the following format: **Group#_InClass01.zip**
   c. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 1 (100 Points)

In this assignment you will practice using Data Structures and Object Oriented concepts in Java. ***Your implementation should target the most efficient algorithms and data structures. You will be graded based on the efficiency of your implementation. You will not be awarded any points if you use simple nested loops to implement the below tasks.*** You should use one or more of the below data structures:

- ArrayList :
  - JavaDoc: http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html
  - Tutorial: http://docs.oracle.com/javase/tutorial/collections/interfaces/list.html
- HashSet :
  - JavaDoc: http://docs.oracle.com/javase/7/docs/api/java/util/HashSet.html
  - Tutorial: http://docs.oracle.com/javase/tutorial/collections/interfaces/set.html
- HashMap :
  - JavaDoc: http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html
  - Tutorial: http://docs.oracle.com/javase/tutorial/collections/interfaces/map.html

Part 1 (50 Points):

You are given the file "*userList1.txt*", which includes user records. Each line of the file represents a single user record. Each record consists of a user's first name, middle initial, last name, age, city, and state. The values are comma separated, for example, Avis,E,Camacho,65,Millburn,NJ.

You are asked to perform the following tasks:

1. PartOne.java should include the implementation for this part.
2. You are also provided with a User class. Simply drag the User.java file provided under your "src" folder in eclipse. Feel free to create any additional classes that are needed.
3. Read the records in the "*userList1.txt*" file. You should implement the parseUser() method in the User class. **Hint**: extract each value from a user's record using Java's String.split() method and set the delimiter to a comma, see provided code below. Each user record should to be assigned to a User object.
4. Your goal is to locate the set of repeated user records in the provided file. Repeated user records are records that appear more than once in the provided file. **Hint**: you need to use one of the data structures listed above.
5. Print the set of repeated user records in ascending order by last name. **Hint**: To sort use the Collections.sort(). http://docs.oracle.com/javase/6/docs/api/java/util/Collections.html

Part 2 (50 Points):

In this part, in addition to the file "*userList1.txt",* you are also given another file "*userList2.txt"*. Both files are formatted in a similar pattern.

You are asked to perform the following tasks:

1. PartTwo.java should include the implementation for this part. You are also provided with a User class. Feel free to create any additional classes that are needed.
2. Your goal is to find the set of users that exist in both files. Users are equal if they have equal attributes. In other words, you should find the set of intersecting records between the two provided files provided.
3. Print the set of overlapping users, sorted in ascending order by age. First print the total number of users who are in the overlapping set, then print the set content in ascending order by age.

The following code reads in a file line by line. It is assumed that the file is included in root folder of the Eclipse project. Use this code to help you read the provided files.

```java
public void readFileAtPath(String filename) {
        // Lets make sure the file path is not empty or null
        if (filename == null || filename.isEmpty()) {
                System.out.println("Invalid File Path");
                return;
        }
        String filePath = System.getProperty("user.dir") + "/" + filename;
        BufferedReader inputStream = null;
        // We need a try catch block so we can handle any potential IO errors
        try {
                try {
                        inputStream = new BufferedReader(new FileReader(filePath));
                        String lineContent = null;
                        // Loop will iterate over each line within the file.
                        // It will stop when no new lines are found.
                        while ((lineContent = inputStream.readLine()) != null) {
                                System.out.println("Found the line: " + lineContent);
                        }
                }
                // Make sure we close the buffered reader.
                finally {
                        if (inputStream != null)
                                inputStream.close();
                }
        } catch (IOException e) {
                e.printStackTrace();
        }
}// end of method
```

String Tokenization:

To split the contents of a single line read a file.

```java
String[] resultingTokens = lineContent.split(",");
for (int i = 0; i < resultingTokens.length; i++){
                System.out.println(resultingTokens [i].trim());
}
```