

ITIS/ITCS 5180 Mobile Application Development
In Class Assignment 01

Basic Instructions:

1. In every file submitted you MUST place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of the student.
2. Each group is required to submit the assignment on Canvas.
3. Please download the support files which include a Java project to be used for this assignment.
4. **Submit Codes:**
 - a. Zip all the project folder to be submitted on canvas.
5. Submission details:
 - a. The file name is very important and should follow the following format:
800#_InClass01.zip
 - b. You should submit the assignment through Canvas: 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>

Question 1 (25 Points)

You are provided with the Data class that contains a users array (Data.users) which is an array of users. Each element in the array represents a single user record. Each record is a string formatted as : firstname, lastname, age,email, gender, city, state. You are asked to perform the following tasks:

1. Your implementation for this question should be included in MainPart1.java file.
2. Create a User class that should parse all the parameters for each user. **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.
3. Your goal is to Print out the TOP 10 oldest users.
4. **Hint:** To sort use the Collections.sort(). <http://docs.oracle.com/javase/6/docs/api/java/util/Collections.html>

Question 2 (25 Points)

Using the same Data.users array. You are asked to perform the following tasks:

1. Your implementation for this question should be included in MainPart2.java file.
2. The goal is to count the number of users living each state. Print out the list of State, Count order in ascending order by count.

Question 3 (20 Points)

In this question you will use the Data.users and Data.otherUsers arrays that include a list of users. You are asked to perform the following tasks:

1. Your implementation for this question should be included in MainPart3.java.

2. The goal is to print out the users that exist in both the Data.users and Data.otherUsers. Two users are equal if all their attributes are equal.
3. Print out the list of users which exist in both Data.users and Data.otherUsers. The printed list of users should be sorted by state in descending order.

Question 4 (30 Points)

You are provided with the Data class that contains an items array (Data.items) which is an array of items in a store. Each element in the array represents a single item record. Each record in the array represents a single item record. Each record is a string formatted as : Name, ID, Price. Also, you are provided with an array called shoppingCart (Data.shoppingCart) which is an array of items' quantities. Each element in the array represents a single item record. Each record is a string formatted as : ID, quantity. You are asked to perform the following tasks:

1. Your implementation for this question should be included in MainPart4.java file.
2. Create a StoreItem class that should parse all the parameters for each item. **Hint:** extract each value from a item's record using Java's String.split() method and set the delimiter to a comma, see provided code below. Each item record should to be assigned to a StoreItem object.
3. Create **the most efficient data structure** that best fit the goal. **Hint:** The selected data structure should facilitate the retrieval of the item details based on the ID.
4. The goal is to print out the receipt bill in the following format:

| ID | Name | Quantity | Price * Quantity |
|-----|----------|----------|------------------|
| 123 | Tomatoes | 10 | \$30 |
| . | | | |
| . | | | |

Total Bill: \$400

Some Sample Code

String Tokenization: to split the contents of a single line read a file.

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