**CSE 274 – Fall 2018**

**PROJECT #1: 70 points – Due Sunday, September 9, by 11:59 p.m.**

**Outcomes:**

* Solve problems involving arrays and arraylists
* Follow a set of specifications for class implementation, including constructors and methods.
* Format and comment source code that adheres to a given set of formatting guidelines
* Test code for edge cases, corner cases, and typical cases

**General description:**

You have been hired by a new social networking startup, PalBook, to develop the core functionality for adding and removing members, allowing members to become "pals", and to perform some analysis about the relationships among members. Software engineers have documented the methods and constructors that need to be implemented. The documentation is provided here: PalBook.html

**Submitting:**

* You will submit one or more Java source code files: **PalBook.java** is required, but you may write and include other Java source code files as needed. If PalBook.java depends on other classes you have written, those other source code files must also be submitted.
* If you name your file, your class, or your methods incorrectly, or if any of your methods or constructors have the wrong parameter types or return types...anything that breaks my tester, you will lose at least 50% of the value of the assignment, and may end up with a score of zero.

**General requirements:**

* Use Javadoc style comments for your PalBook class and any other supporting classes that are meant for reuse. There is no need to use Javadoc style comments for a tester class.
* **At the top of the comments of PalBook, include the following (failure to include this will result in a 5-point deduction):**
  + A section listing which parts of your program work correctly
  + A section listing which parts are not working correctly. For those, explain what's wrong.
* Format your code so that it is readable using generally accepted guidelines for formatting source code.
* Follow good programming practices. Don't make code more complicated than it needs to be. Break larger methods into smaller parts. A method should not be longer than 25-30 lines of code.

**Requirements for the PalBook class:**

* Implement the methods and constructors described here: PalBook.html. Do not create any other public methods or constructors than those specified here. Do not add any public data members.
* You may add other private methods, constructors, and data as needed.

**Ideas to help you:**

1. Consider the following order. Write and test...
   1. constructors
   2. methods that add, contains, getMemberCount and remove members
   3. methods that make two people pals, check if two people are pals, and that end the friendship between two pals
   4. getMembers and getPals (making sure that you are sorting results)
   5. toString
   6. commonPals
2. Ordinarily, you would need to create your own tester class from scratch that uses each of your constructors and methods. I have provided an incomplete tester class for you (see the Google Drive folder). You should add as much code to it as you need, in order to thoroughly test your code.

**Scoring:**

|  |  |
| --- | --- |
| **Requirement** | **Max score** |
| Constructors, addMember, removeMember, containsMember, makePals, endPals, arePals, getMemberCount | 40 |
| getMembers, getPals, toString | 15 |
| commonPals | 5 |
| Handle edge cases, making sure that your methods behave correctly when (for example), you try to add a member to a full network, or you try to remove a member that doesn't exist, or you try to make two people pals but one of them isn't a member. | 5 |
| Use Javadoc comments to document the PalBook class, constructors, and methods. If you've never written Javadoc comments before, here is an 8-minute tutorial that can help you understand: https://www.youtube.com/watch?v=6XoVf4x-tag | 5 |

Sample output of the Tester.java

An empty network:

Add Manar and her siblings:

Kasem[]

Maher[]

Majdi[]

Manal[]

Manar[]

So far, there are no pals in the network:

Manar's pals: []

Give Manar a couple pals:

Manar's pals: [Kasem, Maher, Manal]

Here's the network so far:

Everything should be sorted by name...

Kasem[Manar]

Maher[Manar]

Majdi[]

Manal[Manar]

Manar[Kasem, Maher, Manal]

Remove Kasem from the network:

Maher[Manar]

Majdi[]

Manal[Manar]

Manar[Maher, Manal]

Remove Maher from Manar's pals:

Maher[]

Majdi[]

Manal[Manar]

Manar[Manal]

There are still some methods to be tested.

Also need to test what happens for edge cases...

Adding when the network is full

Removing a non-existent member

Removing when the network is empty

Making pals when one or more names is not a member

And so on...