CT326 Lab 3:

(Note that this is a lab exercise and in <u>not</u> graded. It is recommended that you follow a test-driven development approach and fully document your code with JavaDoc but these aren't necessary.)

a) Write a Java class called `UserAccount` that has the following class attributes: `long userID`, `String name`, `String emailAddress`

Write a Java program that loads up a list of user accounts from a CSV file called users.csv Each line of the users.csv file will look something like this:

29301001122, Tim Stewart, tim@timestewart.co.uk

For testing purposes, create a users.csv file with at least 10 users

b) The natural order for the `UserAccount` class should be based on the `emailAddress` attribute.

c) Define equality for the `UserAccount` class as being true if the `userID` is the same.

d) Implement the `hashcode()` method for the `UserAccount` to provide a hash of the `userID`.

e) Implement an appropriate `toString()` method for the `UserAccount` class that returns a *formatted* string representation of a user account.

f) Then perform the following tests:

1. Use the `Collections.sort()` method to sort the list of users based on Natural Order and print out the result.
2. Sort the list of users again based on the `userID`, in ascending order, using a Comparator defined as an anonymous inner class. Print out the results.
3. Sort the list of users again based on the name, in descending alphabetical order, using a Comparator defined as a lambda function. Print out the results.
4. Use the `Collections.binarySearch()` method to search for a `UserAccount` in the list that has been pre-sorted using the natural order.

g) Create a `Workspace` class with the following class attributes, a method to add a collaborator, and a `toString` method that returns a string including the workspace name and collaborators.

```
String workspaceName, String workspaceDescription,
UserAccount owner, List<UserAccount> collaborators
```

Add the list of `UserAccounts` to a `Map` where the instance of `UserAccount` is used as the key and a `List` of `Workspace` objects is the value. Add at least one workspace for the `UserAccount` at position 0 in the original list, where the workspace has two collaborators (the `UserAccounts` at position 4 and 7 from the original list). Then

retrieve the Workspaces (for the `UserAccount` at position 0 in the original list) from the `Map` using an appropriate key and print out the result.