Learning Outcomes

Upon completion of this lab, you should be able to do the following in Eclipse:
- Add to the classes defined in Topic 1 of the lecture notes.
- Create Java class documentation using the tool Javadoc.
- Use the classes defined in Topic 1 of the lecture notes in an application.

General Lab Instructions to Help Labs Run Smoothly

- Read through the lab instructions before coming to the lab.
- Do the pre-lab preparation.

Preparation. Make sure you understand the parts of a class definition: instance variable declaration, constructor definition, method definition. Read this introduction to Javadoc.

IMPORTANT: Make sure you attend your lab session and sign an attendance sheet, show the result(s) of each exercise to your TA, and submit your .java files through OWL by 11:55pm on the same day as your lab session to get your marks for the lab.

Exercise 1: Adding Methods to the SocialNetwork Class

- Add the following two methods to the SocialNetwork class:
  - `getNumFriends()`: it will return the number of friends in the list.
  - `clearFriends()`: it will remove all friends from the list. (Hint: this can be implemented with just one assignment statement. The SocialNetwork class has an instance variable called `numFriends` that holds the current number of persons in the list of friends. If you set this to 0 (zero), you do not need to clear out the array `friendList` - why not?)
- Add a test harness to SocialNetwork.java to test your new methods by doing the following. (Note that this is not a complete test of the class; it will just give you a start in using a test harness in the Lab. You can add more tests later if you wish.)
  - Create a SocialNetwork object using the constructor `SocialNetwork()`.
  - Add three friends to your social network (use method `add()`).
  - Test the `getNumFriends()` method by printing the value that this method returns.
  - Test the `clearFriends()` method (how will you show that this worked?)
Exercise 2: Using Javadoc comments

- Note that the existing comments in SocialNetwork.java are Javadoc comments. (You should have read the Introduction to Javadoc document before this Lab.) Add Javadoc comments for the new methods you added to SocialNetwork.java in Exercise 1.
- Set up Eclipse to generate documentation for the classes in this Lab, from your Javadoc comments:
  - Go to Project, Generate Javadoc...
  - For the Javadoc command, click on the Configure button and locate the "javadoc.exe" program. This will be in the "bin" directory of the Java Development Kit installation on the machine. This could for instance be:
    * C:\Program Files\Java\jdk1.5.0_03\bin\javadoc.exe
  - Select the project for which you want to generate the documentation.
  - Ensure that Public and Use Standard Doclet are selected, and Destination is the default ({your drive}:{your workspace}:{your project}\doc) and click Finish.
  - Javadoc generates a set of html files which comprise the documentation for the program. You should see these in the Package Explorer in Eclipse under the doc directory. You can view any of the html files by double clicking on it (start with index.html).

Exercise 3: Using the SocialNetwork Class in an Application

- You will now use the SocialNetwork class in an application program. Download MyFriends.java and run it. It is a simple application program that creates and prints a list of friends.
- Add statements to MyFriends to do the following:
  - Call the remove() method to remove one of the friends whose name is already in the list. If the method returns the value true then print a message giving the name of the person that was removed from the list and indicating that the person was removed, for example: "Mickey Mouse was removed successfully." If the method returns the value false then print a message indicating that the specified person could not be removed, for example "Donald Duck was not in the list of friends".
  - Call the remove() method for a person whose name is not in the list of friends, and print a similar message as above.