

04. Fourth Assignment – Facebook Friends

We'll continue with the Facebook application by enabling the ability to friend and de-friend people through our driver program. In particular, you will need to add four new menu options: friend someone, de-friend someone, list friends, and recommend new friends. Of course, you'll need to add new methods to the Facebook class to perform these actions.

The first three of these new functions (friending, de-friending, and listing friends) are similar to the methods we wrote in 03. Third Assignment Binary File I/O – you should first prompt for the username of the FacebookUser doing the action. If there is no user with that username, display an error message. If the user exists, prompt for a password and make sure that what is entered matches the user's password. For friending and defriending, you'll also need to prompt for the username of the new or former friend. Again, if there is no user with that username, display an error message, otherwise, call the appropriate method for the FacebookUser object, passing it the object that represents the new/former friend.

How should we go about recommending new friends? We'll take the approach of recommending all of our friends' friends and our friends' friends' friends, and so on. This is a perfect opportunity to use recursion – we can create a `getRecommendations` method that takes a FacebookUser as an argument. The method should return an ArrayList that contains all of the friends of the FacebookUser that is passed into it *plus* the result of calling the same `getRecommendations` method on all of that FacebookUser's friends. Be careful not to add anyone to the list of recommendations if they are already on it – that could lead to an infinite loop.

You will be graded according to the following rubric (each item is worth one point):

- The driver menu contains the four new options
- It is possible to add a friend at runtime
- It is possible to remove a friend at runtime
- It is possible to list a user's friends
- It is possible to get friend recommendations, and those recommendations are correct according to the requirements given above (i.e. they are all the friends of existing friends, and so on, without duplicates)
- The `getRecommendations` method is recursive (usually assignments will not specify the approach you should take, but this is a special case since this topic is all about recursion)
- Error messages are displayed for invalid usernames or passwords
- The program compiles
- The program runs
- The program is clearly written and follows standard coding conventions

- **Note:** *If your program does not compile, you will receive a score of 0 on the entire assignment*
- **Note:** *If your program compiles but does not run, you will receive a score of 0 on the entire assignment*
- **Note:** *If your Eclipse project is not exported and uploaded to the eLearn drop box correctly, you will receive a score of 0 on the entire assignment*
- **Note:** *If you do not submit code that solves the problem for this particular assignment, you will not receive any points for the program's compiling, the program's running, or following standard coding conventions.*