CS161: Introduction to Computer Science
Homework Assignment 8
Due: 3/28 by 11:59pm

To Do Lists

In this assignment, you will implement a To-Do List application similar to the `BookCase` example we completed in class.

There are 3 Java classes for this assignment: `ToDoItem`, `ToDoList`, and `Controller`.

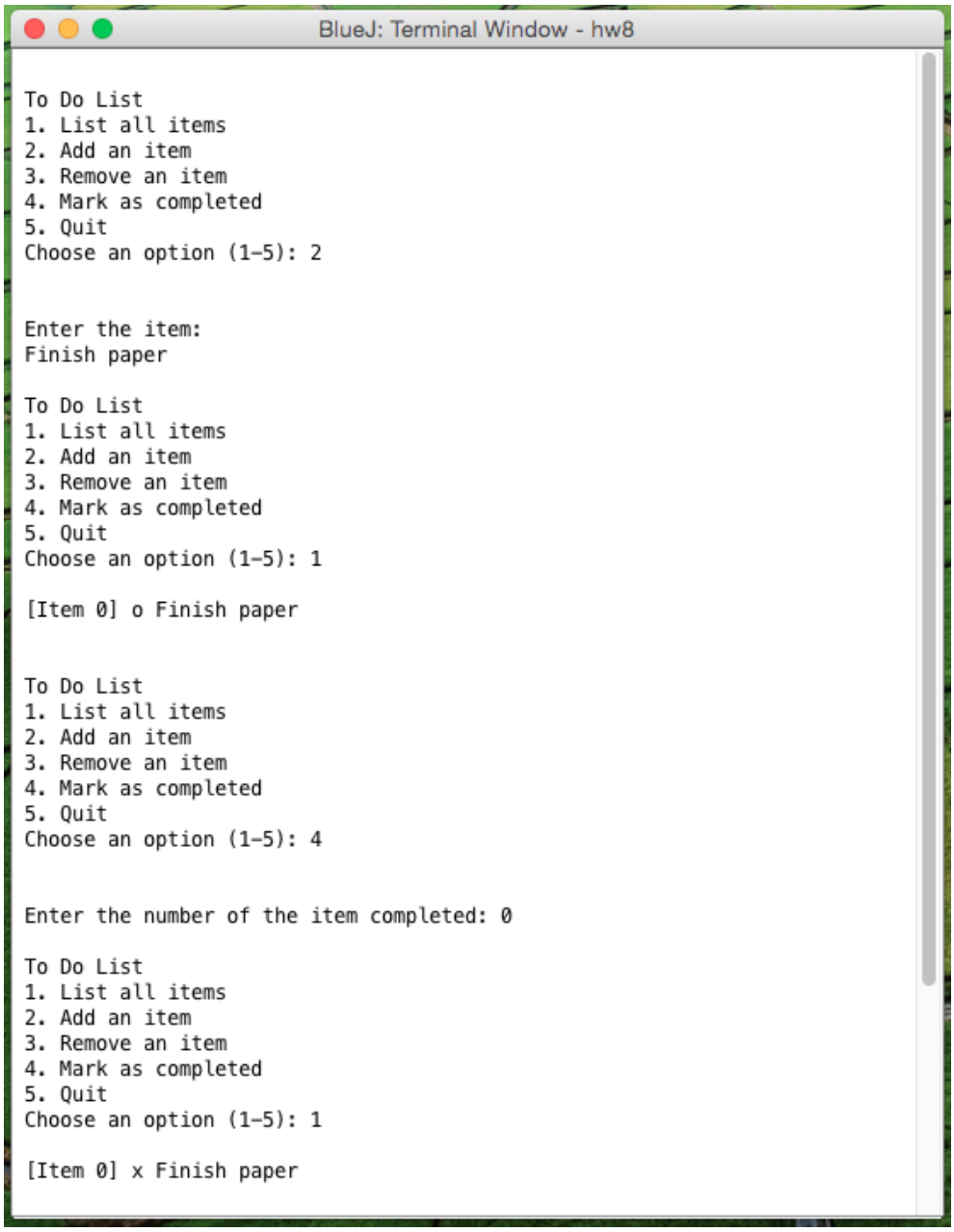
1. The `ToDoItem` class represents a single item on the to do list. This class should contain a piece of text. You should also keep track of whether the item has been completed or not. Add whatever methods to this class that you think are necessary.
2. The `ToDoList` class represents a list of to-do items. In addition to a constructor, the `ToDoList` class should have the following methods:

   - An `addItem(String message)` method that adds a new item to the list
   - A `removeItem(int index)` method that removes an item from the list
   - A `markAsCompleted(int index)` method that marks a given item as completed
   - A `toString()` method that returns a `String` representation of the list

   Note that the `removeItem` and `markAsCompleted` methods take an integer representing an index into the list. You should check that the index passed in is, indeed, a valid index into the list.
3. The `Controller` class should use a do-while loop and a switch statement to continuously print a menu of options for the user to choose from. You can use the `Controller` class from the book case application as an example of how to write this class.

The only print statements from your program should come from the `Controller` *class.* In particular, the `ToDoItem` and `ToDoList` classes should not print anything to the screen.

An example interaction with my To Do List application is shown on the next page:



Note that I used 'o' and 'x' to indicate whether an item has been completed or not. Feel free to get creative and come up with your own design for displaying information to the user!

Extensions

Looking for additional challenges? Here are some interesting ways that you could extend this application:

1. Make a more realistic `ToDoItem` class. For example, you could allow the user to specify a priority of low, medium, or high for each item on the list.
2. Add methods to the `ToDoList` class that allow the user to promote or demote an item on the list – i.e. to move an item up by one or down by one on the list. This would allow the user to order the list according to importance
3. Add methods to the `ToDoList` class that let the user print portions of the list – e.g., print all items that have high priority or print all items whose note contains a certain word. For example, I might want to print all the items on the list that contain the word "cs161" to see what I need to do for this course.
4. A surprisingly difficult thing to do is to remove all completed tasks from the list. This would require a loop to iterate over all items in the list. However, as you're iterating over the list you are also changing the list (i.e. removing items from the list). This is known as *concurrent modification* – you are concurrently traversing and deleting from the list. If you're interested in how to implement this functionality, let me know and I can give you some hints.

Style Guide

Finally, before you submit your assignment, go through the checklist below and make sure your code conforms to the style guide.

Checklist

- ☐ All unused variables are deleted
- ☐ All instance variables are used in more than one method (if not, make them local)
- ☐ Javadoc comment for all classes
- ☐ All methods have Javadoc comments (except for the `main` method)
- ☐ All numbers have been replaced with constants (i.e. no magic numbers)
- ☐ Proper capitalization of variables, methods, and classes
- ☐ Use white space to separate different sections of your code

Read the "Style Guide" (under "Resources" on the course website) for more information.

Submitting your lab assignment

You should submit your `hw8` folder with your `ToDoItem`, `ToDoList`, and `Controller` class inside. Please rename your folder with both your first and last name *before* you zip it!