

Assignment 2: ArrayList

Due: Week five. Check the course web site for the exact due date and time, which are campus-specific.

Web stuff: See the course web site for starter code, hints, and announcements about the assignment.

Purpose: To give practise writing precise comments, and more exposure to interfaces and exceptions. Also to give practise using the memory model.

Overview: You will be given the following code:

- An interface called **Keyed**. Objects that are keyed have an integer value (a “key”) that uniquely identifies them.
- An interface called **List**. Elements of a **List** must be keyed, and a **List** must not have two entries with the same key.
- A class called **ArrayList** that implements **List** using an array. This class uses a trick to save time when deleting: just mark the entry as deleted, rather than actually removing it. This is like crossing a word out from a paragraph on paper; the word is still there, but you know that it’s not part of the paragraph.
- A small class **CalEntry** that implements **Keyed**. Instances of **CalEntry** can thus be put into an **ArrayList**.

There is a **main()** inside **ArrayList** that gives a small example of using an **ArrayList**.

Your task: You can do these in any order, but we suggest the following:

1. Trace in detail, using the memory model, exactly what happens when one executes **ArrayList** up to the comment marked “For question 1, trace everything up to this point.”
2. Write precise and thorough comments for classes **ArrayList** and **CalEntry**. Refer to the section of the lecture notes called “Design by Contract” for detailed instructions on writing good comments; we will compare your work to the high standard set there.
3. As you look at the code, you will find that some of the variable names are horrendous. Improve any variable names in classes **ArrayList** and **CalEntry** that are poor. Refer to the document “Java Guidelines” in the course handbook for guidance on good variable names.

To help us find the name changes you’ve made, fill out the tables that appear in the comments at the top of classes **ArrayList** and **CalEntry**.

What to submit: There are two parts to hand in:

- Submit **ArrayList.java** and **CalEntry.java** electronically. Details on how to do so, including campus-specific requirements regarding file names and submit location will be provided on the different campus web sites.
- Submit your memory model trace on paper, with the assignment cover sheet (available on the course web site) stapled to it. Details on how and where to hand in your paper submission will also be provided on the different campus web sites.

If you hand in the two parts at different times, the *later* of these will be considered your assignment submission time.