

**Name:** \_\_\_\_\_

This homework is *optional*, but highly recommended. It may be used in two ways, which are mutually exclusive. First, those who submit this homework will have their *lowest homework grade dropped*. Second, this homework may be used as a *seed for the term project* by implementing what is described below and extending that work with a variety of additional features.

Please refer to the previous assignment for important instructions on the allowable use of resources and the requirements for electronic and hardcopy submission. Please monitor and use the class newsgroup for questions, clarifications, hints, and other discussions.

This assignment asks you to build an enhanced version of the simple record manager application of Homeworks 2 and 3, in effect transforming it into a simple database for contact information. You are encouraged to build on your code from earlier assignments, but are not required to do so. If you prefer, you may use suitable implementations from the standard Java Collections library instead of your earlier implementation that was based on a level-wise k-search-tree. There are three main enhancements to the earlier application:

**Additional fields** Records are now composed of multiple key and data fields, instead of the simple (key, data) scheme of the earlier version. You are encouraged to add a variety of fields that are useful in a database of contact information, but must include at least last name, first name, and phone number.

**Persistence of data** In the earlier version, all data is lost when the application terminates, a behavior unsuited to our current application. Data must now be stored in a file so that it persists between invocations of the application program. When the application starts, it must automatically load data from the saved file if that file is found; before the application terminates, it must save updated data to the file.

**Graphical interface** Instead of the earlier text-based (standard input/output) interface, the application now sports a graphical user interface that exposes all the functions of the earlier version, as well as the above enhancements, in a suitable graphical form based on the Java Swing library.<sup>1</sup>

---

<sup>1</sup>Mark Allen Weiss, *Data Structures and Problem Solving Using Java*, 3rd edition (Addison-Wesley, 2006), Appendix B.