For this assignment, you will write an “Address Book” applet that keeps track of some contact information for a group of people. Your applet should look similar to the following.

On the right side of the applet is a List of names of people in your address book, in alphabetical order. Clicking on a name fills the text fields to the left with the contact information for that person. To add a new person to the address book, the user can fill in the text fields on the left and press the Insert button. This will insert the new person into the list on the right, in alphabetical order. To delete a person from the address book, the user can click on a name in the list and press the Delete button. This will remove the name from the list to the right, while maintaining the alphabetical order of the list.

There is a working applet on my web page that you should execute to see this in action.

The address book entries (name, e-mail, phone) should be stored in an array of Person objects, where Person is a class you will write to store information about one person in your address book.

Obviously, you will need to implement a sorting algorithm to maintain the array in alphabetical order by name.

When you insert or delete a name from the list, you will need to re-sort the array, clear the list, and repopulate the list with the sorted data. You will need to review List methods in your book to do this.

When an item is selected in the List, and you detect this in your ItemStateChanged(ItemEvent e) method, you can get the selected item with e.getItem(). This will return a generic Object which, when converted to a usable type, will be an integer value signifying the position in the list (starting at 0) of the selected item. For example, in the picture above, selecting “John Kerry” will return a value of 1. To convert the Object returned by e.getItem() to an integer, typecast it first to an Integer object and then typecast the Integer object to an int. This is somewhat similar to what we had to do with the calculator class earlier in the semester.

Note that this is not a “real” address book in the sense that data is not persistent over multiple executions. This would involve working with files, which we will not get to this semester.

It is very important that you follow the methodical programming practices that we have talked about in class. Think very carefully about how you will structure your applet, and what classes you will need, before you ever sit down at a computer to start programming. As issues arise in your planning, resolve them right away so that you have a clear idea of your plan before you start writing code. Once you have a clear plan, write your classes (with constructor/init methods) first, then write outlines of your methods, and then finally write your code. Remember to always stay in control of your project by compiling and testing after every step.

Be sure to include appropriate comments in your code and to test your programs completely before handing them in.

To hand in your code, you should e-mail me your completed project. To do this, locate the folder containing your BlueJ project in the Finder and select it. Then hold down the Control key while you press the mouse button. In the popup menu that appears, there will be a “Create Archive of…” option. Selecting this option will create a file in the same directory with .zip extension. This is a compressed version of your project folder in “Zip” format. Attach this file to an email and send it to havill@denison.edu with Webmail.