Computing and Algorithms II: Programming Assignment 5

CS-102 **Fall 2008**

In this programming assignment, you will demonstrate your knowledge of graphical user interfaces by revising the airport abbreviation code program from Programming Assignment 5 to incorporate a graphical user interface.

New Requirements

For this assignment, you should add a graphical user interface to your airport abbreviation code program. All user input and program output should be conducted through a graphical interface. (Thus, you should never need to use the `System.in` and `System.out` objects, except for debugging purposes.)

All previous functionality in Programming Assignment 4 should be available and functional through the graphical user interface. It should be possible to perform any and every function possible in Programming Assignment 4 in your new version.

You may use any of the standard Java AWT components (including Swing), in any combination you wish, in order to implement your GUI. You should use the Java 1.1 event model (*i.e.* the one presented in class) to interface these components with your application code. Your program should be an application, *not* an applet.

Your program must use the event-driven programming model (*i.e.* components, events, and listeners). It is technically possible to write a GUI program with certain Swing components which never uses events (*e.g.* by using JOptionPanes and nothing else); this is *explicitly* forbidden. (Note that you are still permitted to use JOptionPanes in appropriate places in your interface.)

Your program will be graded not only on correctness, but on the appropriateness of your design and its ease of use. It should be fairly obvious to a semi-intelligent person (such as your instructor) how to use the interface: which buttons perform which actions, which fields contain which information, and so on. This does *not* mean that your GUI needs to be artistically beautiful (indeed, sometimes beautiful GUIs are terribly difficult to use). However, it does mean that components should be layed out in a manner that allows a reasonable user to figure out how to use the program with a minimum of effort.

Submitting Your Program

Before 11:59:59 p.m., Monday, 15 December 2008 (11th Monday), you must email a MIME-encoded email message to `jhuggins@kettering.edu` containing all source code files for your program, including a class named `Prog5` containing a `main` method.

In addition, you must deliver to the instructor a printout of your program files at the start of class on Tuesday, 16 December 2008 (11th Tuesday).

Notes

1. ***Start Early!*** The event-driven programming model is very different from conventional sequential programming, and it may take you awhile to understand how to organize your program. Amidst all your final projects, *please* set aside some time to work on this early!

2. Much of the information you need about the Java AWT is distributed through the Java API; take some time to look around and explore. You may be pleasantly surprised at some of the standard Java AWT components which are available to you, including some which provide some of the functionality required for this program ...

3. *Debugging Hint:* While your program should never need to use the `System.out` object for its operation, you may wish to use it for debugging purposes. For example, it may be very handy to place `println()` statements inside your listener classes, so that you can tell when events are being generated and who is handling them. Alternatively, you may want to leave some of your old print methods in place, so that you can compare the text-based output with your GUI-oriented output. (Of course, these statements should be commented out or removed when your program is complete.)