Lab Exercise #5 – Event Handling
Computer Science 2334
Due by: Friday, March 12, 2010, 5:30 pm

Members: _____

               _____

               _____

               _____

               _____

               _____

Objectives:

1. To learn how to use event handling in Java by completing an information display program.
2. To learn how to modify a class such that it implements and uses **ActionListener** as the event listener on Graphical User Interface (GUI) components.
3. To learn how to implement `actionPerformed()` to handle multiple events from components.
4. To see how to use a **LinkedHashMap**.
5. To demonstrate this knowledge by completing a series of exercises.

Instructions:

This lab exercise requires a laptop with an Internet connection. Once you have completed the exercises in this document, your team will submit it for grading.

Make sure you read this lab description and look at all of the source code posted on the class website for this lab exercise before you begin working.

Assignment:

Event handling is an important feature of Java; one that could be used in a future project. Carefully inspect how it works and the documentation comments included in the code.

1. Download the `Lab5-eclipse.zip` project archive from the class website.   Import the project into your Eclipse workspace using the slides from Lab #2. You will submit the modified project archive when you are finished

2. Review the source code for the **OlympicTeam** class, which is a class for representing (some aspects of) Olympic teams. It is extended by the **OlympicTeamModel** class. This class is the data model for the program. It extends the **OlympicTeam** class by adding variables and methods, and by overriding methods, in order to deal with the GUI. You will use methods provided in these classes to complete the code for the lab.

3. Review the source code for the **OlympicTeamInputWindow** class, which is a class that presents a GUI window to the user for adding new athletes to a team's roster or for clearing out the roster. Label each component of the GUI input window below with the corresponding code variables from **OlympicTeamInputWindow**.

4. Review the source code for the **OlympicTeamDisplayWindow** class, which is a class that presents a GUI window to the user to view a team's roster. Label each component of the GUI input window below with the corresponding code variables from **OlympicTeamDisplayWindow**.

5. Register each object that will serve as an **ActionListener** with its corresponding GUI input component. This means that you need to "connect" the **ActionListener** objects to the objects that are listening for events.

6. Add an `actionPerformed()` method to each class that serves as an **ActionListener** (the classes that implement the **ActionListener** interface). This method should handle the events specified in the source code of **Driver**.

7. Ensure that there are no warnings generated for your code. **Do not suppress warnings.** Fix your code so that warnings are not necessary. (If you can't figure out how to fix your code to avoid the cast warning on the cloned `actionListenerList`, you may leave in that warning.)

8. Submit the **project archive** following the steps given in the **Submission Instructions** by **March 12, 5:30pm** through D2L (http://learn.ou.edu).

9. Turn in this lab handout (with completed answers) to your lab instructor during lab hours or to Professor Hougen during his office hours (4:30-5:30) on February 12 or earlier by bringing it to his office and handing it to him or sliding it under his office door if he is not available.