**Data Structures in Java Lab 4**

In this lab you will write a program to keep track of Olympic athletes. Each athlete represents their home country and may compete in several events in the Olympics. In each event the athlete has a chance to win a gold medal (if the athlete comes in first), a silver medal (if the athlete comes in second), or a bronze medal (if the athlete comes in third).

**Athlete**

The program will use an Athlete object to hold the information about an athlete. An Athlete object will contain the following information for each athlete: the athlete's first name, last name, country, sport, event, and their final result in the event (finished 1st, 2nd, etc.). Athletes who compete in more than one event will be represented by one Athlete object for each of his or her events.

As always, I want you to avoid set and get methods. If another class needs to access an Athlete field, create a method that will query that field. For example, if you need to know an athlete's country, create a boolean method with a country name as the parm. This method will return true if that is the country of the given athlete. Athlete will also need an equals method and it will need to implement the Comparable Interface and have a compareTo method that will be able to compare Athlete objects; Athletes will be ordered by country, and by name (last name and then first name) within country. The ordering established by the compareTo method is called the "natural order" for the class. Remember we found in our class exercise that it is essential to correctly override the equals method inherited from class Object.

Make sure that Athlete objects display in a nicely formatted, readable manner. When you print an Athlete, print the name of the medal if the Athlete came in first, second, or third. Otherwise, print the result number achieved in the event.

**Olympic Athlete**

The class OlympicAthletes will hold the information about all athletes that competed in a given Olympics. The athlete information in an OlympicAthletes object will be in a sorted list contained in the OlympicAthletes object. This sorted list will contain the Athlete objects. The Athletes will be ordered using the natural order defined by the compareTo method for Athletes: by country, and by name (last name then first name) within country. You will use the Sorted Array Based List covered in Chapter 6 of your text to build, modify and use the sorted list.

OlympicAthletes will need methods for the following:

* Add an athlete
* Delete an athlete
* Print all athletes
* Print a particular athlete: given the name and country of the athlete, print all the entries for that athlete.
* Print an event: given the name of the event, print all athletes that competed in that event.
* Print medal winners by country: given the name of the country, print all the athletes from that country that won medals.
* Print medal winners by event: given the name of the event, print the medal winners for that event.
* Print medal winners by medal: given the name of the medal, print all athletes that won that medal.
* Print all Athletes in the List in alphabetical order by Athlete name (last name, first name). Hint: you need to add a comparator to the Athlete class as this is not the natural order.
* Print all Athletes in the List in alphabetical order by event. Hint: you need to add a comparator to the Athlete class as this is not the natural order.

**Client Code**

Your main program will create an OlympicAthletes object for the Summer 2016 games that were held in Rio or the Winter 2018 games that were held in Pyeongchang. The data for the Athletes will be read from a file. **Your file must use the following format**: the data for each athlete must be entered in the following order: first name, last name, country, sport, event, and result, with each of these on a separate line. You can assume that the data in the input file is valid, and your program should read till end of file (do not use a sentinel). For example, a very small data file would look like:

Jamie

Anderson

USA

Snowboarding

Women's Big Air

2

Ireen

Wust

The Netherlands

Speedskating

Women's 3000 Meter

2

Miho

Takagi

Japan

Speedskating

Women's 3000 Meter

5

Laurie

Blouin

Canada

Snowboarding

Women's Slopestyle

2

Lotte

Van Beek

The Netherlands

Speedskating

Women's 1500 Meter

4

Your client code should read the input file and store the data in your OlympicAthletes object, then give the user a menu so they can print all athletes, look up athletes, look up events, look up medal winners (in the ways described above), add athletes, and delete athletes. Let the user continue until they want to quit. If the user gives an invalid response to the menu, give an error message and continue.

**Do not put all of your client code in one or two functions. Use a function for each logical part of the processing.**

Your input file should have at least 30 athletes. You can obtain Olympic information by going to [olympics.com](http://www.olympics.com/) or go to your favorite search engine and perform a search on the Summer 2016 games or the Winter 2018 games. For the winter games I found [ESPN's 2018 Winter Olympics Results](http://www.espn.com/olympics/winter/2018/results) (http://www.espn.com/olympics/winter/2018/results) easy to use.