CS-1181 Lab Problem 4: Most Valuable Player

PURPOSE: To review classes and interfaces that use generics

DIRECTIONS:

Part A (Due by end of first lab session)

Your task is to develop an `Athlete` class that has the person's name and sport where the name and sport are strings. Your `Athlete` class should implement the `Comparable` interface. When comparing two athletes, they should be sorted alphabetically by sport. If two athletes have the same sport, they should be sorted alphabetically by name.

In your main method, create an `ArrayList` of five athletes. Print the list, then sort the list using `Collections.sort()` and print the sorted list.

You may need to add getter methods for the fields in your `Athlete` class, and it might be helpful to override the `toString()` method to make it easier to display the object.

EXAMPLE:

```
Unsorted
John Doe (baseball)
Sam Johnson (football)
Kevin Smith (baseball)
Sally Johnson (swimming)
James Smith (swimming)
Meagan Kelly (swimming)

Sorted by sport and then name
John Doe (baseball)
Kevin Smith (baseball)
Sam Johnson (football)
James Smith (swimming)
Meagan Kelly (swimming)
Sally Johnson (swimming)
```

Part B

Update your `Athlete` class to include an `Athlete`'s worldwide rank in their sport (first, third, 105th, etc.). The rank can be **anything that is comparable** (do not hard-code this to int - use generics). Update the `toString()` method for class `Athlete` to also display an `Athlete`'s rank.

Update your main method to do all the same things as in Part A, and then to print the list sorted by sport and then by rank. That is, if two athletes have the same sport, they should be sorted in increasing order of rank.

For the first sort, you will use the built-in `compareTo()` method of class `Athlete` **(that is, just call `Collections.sort()`, as before)**. For the second sort, you will pass a `Comparator<Athlete>` object to `Collections.sort()`. That is:

- First (with `compareTo()`): Sort alphabetically by sport. If two athletes have the same sport, then sort them alphabetically by their name.
- Second (with `Comparator<Athlete>`): Sort alphabetically by sport. If two athletes have the same sport, they should be sorted in decreasing order of rank.

  For full credit, ensure that your program is well commented and follows JavaDoc standards for your method(s). Comments are only required for the Part B segment of the lab.

EXAMPLE:

```
Unsorted
John Doe (baseball - 3)
Sam Johnson (football - 2)
Kevin Smith (baseball - 1)
Sally Johnson (swimming - 3)
James Smith (swimming - 4)
Meagan Kelly (swimming - 1)

Sorted by sport and then name
John Doe (baseball - 3)
Kevin Smith (baseball - 1)
Sam Johnson (football - 2)
James Smith (swimming - 4)
Meagan Kelly (swimming - 1)
Sally Johnson (swimming - 3)

Sorted by sport and then ranking
Kevin Smith (baseball - 1)
John Doe (baseball - 3)
Sam Johnson (football - 2)
Meagan Kelly (swimming - 1)
Sally Johnson (swimming - 3)
James Smith (swimming - 4)
```

RUBRIC:

[1pt]	**Documentation**
[1pt]	**Part A correct**
[1pt]	**Part B correct**