CSC207: The Comparable Interface

To earn the credit for this lab, you must be present in the lab class. For this lab, and for the rest of the labs in this class, please make sure to submit by end of the lab class.

1 Overview

This week, you are going to practice interfaces and in particular, the Comparable interface. In this lab, you are encouraged to work in pairs. You and your partner together will be able to figure out problems better than you would individually. Find yourself a partner. If you have trouble finding one, let your TA help you. This partnership is only for today's lab. We strongly advise you to form a partnership with a colleague who has a similar level of background.

2 Choose a driver and a navigator

In all the labs, we will use the terms *driver* and *navigator*. Here are the definitions of the two roles:

- **Driver**: Types at the keyboard. Focuses on the immediate task at hand.
- **Navigator**: Thinks ahead and watches for mistakes.

In lab handouts, we'll often refer to you as `s1` and `s2`, and `s1` will be the first driver.

3 Log in and get things set up.

Use MarkUs to form a group with your lab partner. The MarkUs item for this lab is Lab4. This will require `s1` to log in to MarkUs and invite `s2` to form a group. Then `s2` will need to log in and accept the invitation from `s1`. Take a note of the group MarkUs created for you. It will be of the form `group_nnnn`.

All repositories in this course will have a URL of the form:
`httsp://markus.cdf.toronto.edu/svn/csc207-2016-05/repo-name`

`s1` drives and `s2` navigates.

Now `s1` logs in and opens a new terminal window.

1. Change to `s1`'s home directory.
2. Check out your group's repository.
3. In your repository, you will find a newly created directory called Lab4.

4 What to do in this lab

1. Open up Eclipse. Point your workspace to the folder `repo-name`.
2. Using Eclipse, create a new Java project called `Lab4`. Make sure you open the Java perpective.
3. Create a new package called `lab2`.
4. Within `lab2`, create a new Java class called `Football` that implements the Comparable interface.

5. Define the class variables as indicated below.
6. Switch the driver and the navigator.
7. Create the class constructor.
8. Override `equals, hashCode` and `compareTo`.

   Switch your roles.
9. Create a new Java class called `FootballData`
10. Make sure to check the `main` method.
11. Declare an array of Football type named `myTeams` with at least 5 elements.
12. Using the data from the following link : `http://footballdatabase.com/ranking/world/1` instantiate all elements of your array.
13. Switch the driver and the navigator.
14. Using a loop, display the elements of your array.
15. Issue the command `Arrays.sort(myTeams);` sort your array.
16. Display the sorted array.
17. Go back to your terminal window. Make sure you change your current directory to your repository. Issue `svn add Lab4/src` and commit.
18. Demo your program to your TA and sign the attendance.

5 The `Football` class

The `Football` class implements `Comparable` and contains the following attributes:

```
String country;

String club;

int points;
```

Next:

- Add a constructor: `public Football(String country, String club, int points)`
- Override `toString()` so it displays your `Football` objects according to this example:

  `Spain, Barcelona, 2025 points`
- Override `equals` method so two `Football` objects are equal if all their attributes have the same value
- Override `hashCode` (it is up to you to decide on the details of this).
- Implement `compareTo` as follows: `f1.compareTo(f2)` returns -1 if the club name of `f1` is alphabetically smaller than the club name of `f2`, returns 0 if he club names are the same, and +1 otherwise.

6 Submitting your work

There is nothing else that you need to do to "submit" this lab. We will simply examine the history and content of your repository when marking.