Practical 12- Three more ways to sort

Comparable Comparator

For this lab you will need your Friend class and your ArrayUtilities class. Copy and paste your ArrayUtilities class into your Friend project

1. Create 5 Friends and edit your bubble sort so that it accepts an array of Friends and sorts on your getLastName() method. Every time you want a different sort order you are going to have to write another overloaded sort method.
2. Instead of writing your own sort we could use one from the Java library. Let’s try one of the java.util.Arrays.sort methods. The first one requires that the Friend class specifies how it should be sorted. It does this by making the Friend class implement an interface called Comparable, (this forces the Friend class to have a method called compareTo(Friend o). Make your compareTo method sort by last name and if the last names are the same then by first name.
3. It may be the case that there are lots of ways that we might want to sort our Friend records, the last technique gets you to make 2 separate sorting classes. Each of these classes implement an interface called Comparator this interface forces your classes to have a method called compare that accepts two objects. Make a class called LastNameCompare and AgeCompare.
4. Make a Comparator class that sorts one of your enum fields.   
     
     
     
   **The ArrayList class**
5. In assignment one you will be reading from a large file of possibly unknown length so instead of an array you could use an ArrayList. You do not need to specify the length of a list you just keep adding to it. It is ok to have two App classes in one project. Note: you do not need to loop to print out the contents of the list, although you can “for each” your way through a list if required.
6. Make an ArrayList to hold your five friends make the List only hold Friends by declaring it as follows:  
   List<Friend> friendsList = new ArrayList<Friend>();
7. Add the five friends into the list and print them to the screen.
8. Sort them by using the Collections.sort method.
   1. By last name
   2. By age
   3. By one other field

Read Chapter 16 from Head First Java

Read <http://www.tutorialspoint.com/java/java_interfaces.htm> for a description of an interface

1. Create a class called Book that contains two string properties Title and Author. Generate getters and setters and a toString. Eclipse will auto generate your getter, setters and toStrings for you.
2. Create the following books:
   * *Rachel's Legacy* Thomas, Julie
   * Bulibasha Ihimaera, Witi
   * ChappyGrace, Patricia
   * Wild Pork and WatercressCrump, Barry
   * The Invisible MileCoventry, David
   * Dad ArtWilkins, Damien
   * The Antipodeans McGee, Greg
   * AbsenceKing, Joanna
3. Create an array list of these books
4. Sort your books by title by implementing Comparable
5. Sort all your books by author by creating a Comparator