CISC275 Comparators

NOTE: As you do this exercise, focus on **understanding** every piece of code you write. Copying code examples without understanding leads to ugly code and bad exam grades.

NOTE2: Today's exercise is disposable. Make everything public and do not write any methods you don't absolutely need.

- Go to:

  `http://docs.oracle.com/javase/tutorial/collections/interfaces/order.html`

  Read about Collections (note the 's'), Comparable, and Comparator. Where are they and what are they for? Answer very briefly in a text file before proceeding. Save the text file to study for the exam[1].

- Write an Animal class with name and number of legs. You will never create an instance of Animal.

  WHY? Make it an interface or abstract class

- Write a Dog class. A dog is an animal. Write a toString() method for this class.

- Create an ArrayList of dogs and put three dogs in it. Two must have the same name, but different numbers of legs. Print the list before going further (hint: if you write a loop you are doing it wrong).

- Write a comparator class AnimalCompare with the required method and the required parameters. The method must consider name first (ignore case), then legs to break a tie.

- Sort the list of dogs using your comparator and print it again. Be sure that your original ordering demonstrates the full power of your comparator.

- Implement Comparable. Show your dogs being sorted by number of name/legs (your Comparator class), then by legs/name (Comparable), using the same class in the same main.

- Answer for yourselves: What is the major benefit of Comparator over Comparable? Which one implements the Strategy design pattern? What are the syntactic differences in the methods used by Comparator, Comparable to compare objects? Which one should you use?

[1] What is the difference between Collection and Collections?