Using the *Comparable* Interface

1. Write a class Compare3 that provides a static method *largest*. Method *largest* should take three *Comparable* parameters and return the largest of the three (so its return type will also be *Comparable*). Recall that method *compareTo* is part of the *Comparable* interface, so *largest* can use the *compareTo* method of its parameters to compare them. See Section 5.? of the text for a description of the *compareTo* method.

2. Write a class Comparisons whose main method tests your *largest* method above.
    - First prompt the user for and read in three strings, use your *largest* method to find the largest of the three strings, and print it out. (It's easiest to put the call to *largest* directly in the call to println.) Note that since *largest* is a static method, you will call it through its class name, e.g., *Compare3.largest(val1, val2, val3)*.
    - Add code to also prompt the user for three integers. Try to use your *largest* method to find the largest of the three integers. What error message do you get if you pass the integers in directly? This is a situation where the Integer wrapper class is useful. Create a new Integer object to hold each of the integers and pass the objects to *largest*. Again, it's easiest if you put the call directly in the call to println.