

1. This is a pair-programming exercise, so please pair yourself with another student. **PART OF THIS EXERCISE WILL BE PRACTICING JUNIT. IF YOU'VE NEVER USED JUNIT BEFORE, BE SURE TO PAIR YOURSELF UP WITH SOMEONE WHO HAS!**

2. Write a class called **Hurricane**, which **implements the Comparable interface**. (Don't forget to use generic notation.) Your class should have three private instance variables:

```
String name;  
int windSpeed; // measured in miles per hour  
int rainfall; // measured in inches per hour
```

- Provide a public constructor which allows the user to specify the name, windSpeed, and rainfall values as arguments.
  - Provide public "getters" for each of the three fields.
  - Write a simple toString method that returns the state of all three variables in some pretty format.
  - The "natural order" for Hurricanes should be based on their windspeeds.
  - Be sure to write an equals method that is compatible with your compareTo method.
3. **Using JUnit 4** (not JUnit 3), write a series of JUnit tests to check if your compareTo and equals methods are working properly.
4. Write two classes implementing the Comparator interface. (These will be used to compare Hurricanes, so be sure to use the proper generic notation.) The first one will be called NameComparator, the second one will be called RainfallComparator. They will order the hurricanes in the obvious way, according to these names.
5. Write more JUnit tests to check if these two comparators are working properly.
6. Write a separate Driver class that does the following:
- a) Instantiate 100 hurricanes with the names "Hurricane1", "Hurricane2"..."Hurricane100". Use a random number generator to initialize the windspeeds to values in the range of 70 to 200; also initialize the rainfall values randomly to values between 1 and 5.
  - b) Print out all the hurricanes three times:
    - i. Sorted by windSpeed
    - ii. Sorted by Name
    - iii. Sorted by Rainfall

(Hint: Use the Collections.Sort method.)