Homework 2

This is an individual homework assignment. You may receive help from other class mates, but you may NOT copy their work. Copied work will be considered plagiarism and dealt with according to the syllabus and university policy.

Objective: For this assignment, you will demonstrate that you can correctly merge together the results of the prior 3 skill assessments into a single functional program while providing for several more technical functions detailed below. This will include following proper naming, formatting and style conventions as well as using several functions and classes in the Java standard library.

Description: Properly defined classes working with various utility functions are the norm for Java applications. Interactivity with the end user, proper data representation and protection in memory, and correct calculations are required of all business applications. You will join together the components from the last 3 skill assessments into a single Java program that collects several data points from the end user for multiple instances of your defined class, processes their input, and outputs the results of your processing. You must use appropriate Java standard library functions and classes for all tasks. An example dialog is exemplified below in Table 1.

Assignment:

- Create a command-line Java program that performs the following:
  - Works with the provided Driver class test harness to ask the end user for a series of 5 grades associated with 3 unique students
    - The test harness provides the names and the mainline logic
    - The end user will provide the actual grade values during runtime
    - These user provided values must be stored into each appropriate Student instance
      - Do not store student grades in incorrect instances
  - You must modify the Student class from skill assignment 3 to incorporate any fixes indicated by the professor, but also to rename the old .describeMe() method to now conform to the .toString() expectation
  - You must also implement sorting functionality by properly using the Comparable interface and the required .compareTo() method

- Upload your .JAVA code file(s) to the appropriate Blackboard assignment

```
Please enter Grace Hopper's grade as a decimal value (e.g. 79.5):
98.5
Please enter Grace Hopper's grade as a decimal value (e.g. 79.5):
96.4
Please enter Grace Hopper's grade as a decimal value (e.g. 79.5):
97.5
Please enter Grace Hopper's grade as a decimal value (e.g. 79.5):
92.6
Please enter Grace Hopper's grade as a decimal value (e.g. 79.5):
91.8
Please enter Steve Jobs's grade as a decimal value (e.g. 79.5):
56.4
```

```
Please enter Steve Jobs's grade as a decimal value (e.g. 79.5):
45.3
Please enter Steve Jobs's grade as a decimal value (e.g. 79.5):
32.1
Please enter Steve Jobs's grade as a decimal value (e.g. 79.5):
0
Please enter Steve Jobs's grade as a decimal value (e.g. 79.5):
0
Please enter Ada Countess of Loveleace's grade as a decimal value
(e.g. 79.5):
85.7
Please enter Ada Countess of Loveleace's grade as a decimal value
(e.g. 79.5):
81.9
Please enter Ada Countess of Loveleace's grade as a decimal value
(e.g. 79.5):
92.3
Please enter Ada Countess of Loveleace's grade as a decimal value
(e.g. 79.5):
88.4
Please enter Ada Countess of Loveleace's grade as a decimal value
(e.g. 79.5):
95.6
Ada Countess of Loveleace's grade average is: 88.78000000000002
Grace Hopper's grade average is: 95.36
Steve Jobs's grade average is: 26.759999999999998
```

Table 1 – Example Input/Output