Community College of Philadelphia

CSCI211 – Data Structures and Algorithms
ASSIGNMENT 1

Instructor: Michael Hackett
Department: Computer Science
Email: mhackett@ccp.edu

Card Sorter

Using C++, Java, or Python, design a program that creates an array of Card objects, shuffles them, then sorts them by value and then sorts them by value **and** suit.

You will need to create a Card class with the following two fields:

- value (a String)
- suit (a String)

You may include any other fields, constructors, or methods as needed.

Your single array will need to contain 52 card objects, one for each value of each suit. The values to use are **2-10, J, Q, K,** and **A**, with A being the card with the highest value. The suit values to use are **H** (for Hearts), **D** (for Diamonds), **C** (for Clubs), and **S** (for Spades).

After building the array, shuffle the array using the Fisher-Yates Algorithm. You must implement this algorithm yourself.

After shuffling, print each object's value and suit to show the array was shuffled. The output for each object printed should be VALUE - SUIT. For example: **2 - H**.

Then, you'll need to sort the array based on the values. All 2 cards should be at the beginning and all A cards should be at the end; The suit is irrelevant here. After sorting, print each object's value and suit again to show the array was sorted based on the cards' values.

- If completing this project in Java, you may use the Comparable interface and/or the Comparator interface and the Arrays.sort() method. If completing this project in C++ or Python, you must choose and implement a sorting algorithm yourself. You may choose any sorting algorithm shown in the Module 1 lecture slides. You can also implement a sorting algorithm yourself if you are using Java.

Finally, you'll sort the array based on the **values and suit**. Clubs should be first, then Diamonds, then Hearts, then Spades. Each suit should be sorted from 2-A. After sorting, print each object's value and suit again to show the array was sorted based on the cards' values and suits.

- If completing this project in Java, you may again use the Comparable interface and/or the Comparator class and the Arrays.sort() method. If completing this project in C++ or Python, you must choose and implement a sorting algorithm yourself. You may choose any sorting algorithm

shown in the Module 1 lecture slides. You can also implement a sorting algorithm yourself if you are using Java. You can use the same algorithm for both sorts, if you wish.

Submit any and all related source code files in the Assignment 1 submission link.

Grading

See Assignment Rubric in Canvas.