CPS 109 Assignment One, Fall 2017

Due date: 11:59pm October 23, 2017

Learning objectives

The learning objectives of this assignment are, as in Big Java:

1. To implement decisions using if statements
2. To write statements using the Boolean primitive data type.
3. To compare strings and/or characters.
4. To write loops using while or for.

Representing playing cards and hands of cards

An individual playing card is represented as a **string of two characters** so that the first character is the rank (from `"23456789TJQKA"`, and note that 10 is encoded as letter `T` to make all card ranks to be single letters) and the second character is the suit (from `"cdhs"` for clubs, diamonds, hearts and spades respectively). For example, `"Jd"` would be the jack of diamonds, and `"4s"` would be the four of spades.

A hand made up of multiple cards is given as string encoding all those cards consecutively, for example `"Kh3h7s8h2h"` for a five-card hand that happens to have one spade and four hearts. **Note that the cards can be listed inside the string in any order, not necessarily sorted by suit or rank.** The suits and ranks are also **case sensitive**, with the rank always given as a digit or an uppercase letter, and the suit always given as a lowercase letter from the four possible letters *cdhs*.

Class and methods to write in this assignment

Write the class `AssignmentOneF17` (the class must be named **exactly that**) whose `main` method repeatedly reads from the console a five card poker hand, given as ten-character string as described above, or the word "quit" to exit the program. Given the hand, the program should identify what kind of poker hand the user enters, and print the type of hand on the console. There are five types of hands that your program should recognize: **pair, three of a kind, flush, full house, and four of a kind**.

Your program can assume that each input is a legal five-card poker hand from one of five possibilities, or the word "quit", so your program does not need to recognize and recover from illegal inputs.

An example run of the program might look like the following, with the user input in italics:

```
Qs7s2s4s5s
flush
7h8hKsTs8s
pair
2h4d2d4s4c
full house
KsKhKc8sKd
four of a kind
3s9hTh9s9d
three of a kind
quit
```

Your program should print only the answers exactly the same way as the previous example, but not any kind of prompt for the human user, to allow our automated testing of your program. Make sure that your program will **print exactly one line of output for each hand that it reads in**, so that its lines of output match those expected by our automated tester.

Hints for recognizing poker hands

In a five card poker hand, note that the ranks of the cards are given in even-numbered positions 0, 2, 4, 6, and 8, and suits are given in odd-numbered positions 1, 3, 5, 7, 9. Use the `String` method `charAt` to extract the character from the given position of the input string.

The easiest type of poker hand to recognize is the flush. Write a loop that checks that all five suit characters are equal. Ranks don't matter in this task, so you can simply ignore them, since for simplicity, we assume that **straight flushes don't exist**, so you don't need to check whether your flush is also a straight.

To recognize the four of a kind, full house, three of a kind and pair, you could write specialized conditions for each of these. Checking for these types of hand shapes, the suits don't matter at all, so you should just look at the ranks of the cards. (You should check for these shapes in the exact order listed here, which allows you to assume in each test that the hand is not the previously checked hand shape, which will simplify your current test.)

However, being a bit more clever, you can use two nested loops to loop through all 10 pairs of cards in the hand, and count how many of these pairs have the same rank for both cards. From this count alone you can recognize the shape of the hand.

Marking scheme

On BrightSpace, submit a single file called **`AssignmentOneF17.java`,** meeting the requirements

specified above. Your assignment submission will be marked out of **13**. Initially your submission will be graded by an automated test script that compiles and runs your code, giving it ten poker hands as input, followed by the word "quit". For each hand that your program recognizes correctly, you receive 1 mark, for a maximum of **10 points**. If your program does not compile without errors, it will receive a zero mark. *Please note that your program output must be exactly as required in the specification for the automated tester to give it any marks.* ***Do not print anything else, just the bare required answer***. After the automated testing, the TA doing the grading will read through your code to examine its style, structure and logic. Your code should be properly commented and properly indented, with all variables named clearly in standard Java style. The structure and logic of the code should not be unnecessarily long or convoluted. The TA will award up to **3 points** for style, structure and logic. Thus, the best possible score will be **13 points**. The submission deadline is firm, no assignments will be accepted after the deadline.

Plagiarism detection

You are to work alone when writing your code. You can discuss general ideas with your classmates, but you cannot copy code or develop code together nor take code from the web. We will be using the Measure of Software Similarity (MOSS) to identify cases of possible plagiarism; see the following link for details: http://theory.stanford.edu/~aiken/moss. Note, MOSS can detect changing identifiers and rearranging code. The Department of Computer Science takes the act of plagiarism very seriously. Those caught plagiarizing (both originators and copiers) will be sanctioned. Please see Ryerson University's Policy 60 for possible penalties and consequences: http://ryerson.ca/senate/policies/pol60_procedures.pdf. If you are unsure what constitutes plagiarism, please see your instructor.