Java Review Exercises

Re-submit Assignment

Due Jan 12, 2017 by 11:59pm **Points** 50 **Submitting** a file upload **File Types** zip

Exercises - Java Review

Answer the following questions by implementing the code samples and/or answering the questions in a word document. Upload all project files and word documents zipped to the exercise drop box.

Part #1

1. Create a class called PlayingCard that stores some basic features of playing cards. Your class should have the following:

- Fields: color, width, height
- Constructors: a parameterized constructor (no default constructor allowed)
- Getters and setters for each field
- An appropriate toString() method

2. Create a child class of your PlayingCard class that can be used in a **Poker Game** (https://en.wikipedia.org/wiki/Poker) . You will need to do a little research on the cards that are present in a Poker game deck. Your child class should have at least the following:

- **Fields:** suit, rank
- **Constructors:** a parameterized constructor (no default constructor allowed)
  - It should not be possible to instantiate a poker card object with invalid suits or ranks (look up the valid values for both these fields)
- **Getters and setters:** for each field
- An appropriate toString() method

3. Create a driver class called CardGameTest. Instantiate an array of poker card objects. Your array should have all cards in a **52 card deck** (https://en.wikipedia.org/wiki/Standard_52-card_deck) . Loop over your array and print out each of the cards using their toString() methods.

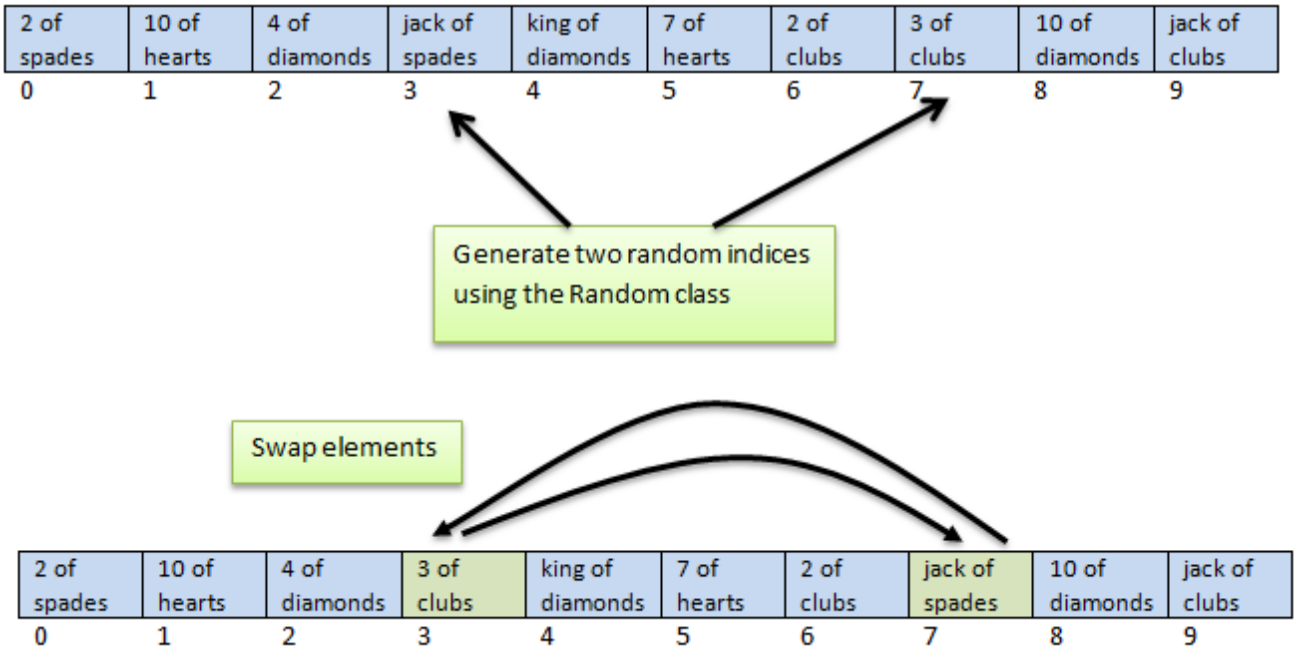
Part #2

If you are feeling rusty with Console IO or the Random class, then please review the following tutorials to get up to speed.

- Review **pdf** on the Random class.
- Review **tutorial** (http://web.eecs.utk.edu/~bvz/cs365/examples/datacheck.html) on Console IO.
- Review another **tutorial** (http://pages.cs.wisc.edu/~hasti/cs368/JavaTutorial/NOTES/JavaIO_Scanner.html) on console IO.

4. Write a method, called shuffle(), that takes your array of Card objects and shuffles the order of elements in the array.

Note: you can approximate shuffling by repeatedly and randomly picking two elements in the array and swapping their position.



Note: Your shuffle() array above should swap different elements several times (pick a large number that makes you confident that most elements are not in their original positions).

5. Create a second method that allows you to shuffle your array of cards n times, using the routine above, where n > 0.

Note: You should have no redundant code in your submission. Don't copy-paste code!

6. Create a method that prints the contents of your Card array to the console. Use this method to test your results from part #1 and #2.