Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

20.6 Program 2b: Card Deck

Objectives

1. Practice with classes and objects
2. Code a class of a deck of playing cards

Background Reading and Resources

- ZyBooks Chapter 7 Objects and Classes
- Make sure you have an IDE that you can use

Program to turn in

Rolling with the same theme, we are going to focus on the characteristics and functionality of a deck of cards. In this program you will be creating a "CardDeck.java" file in your IDEs that will have the following characteristics:

Class Variables:

```
public String[] ranks = {"2", "3", "4", "5", "6", "7", "8", "9", "10",
"Jack", "Queen", "King", "Ace"};
public String[] suits = {"Clubs", "Diamonds", "Hearts", "Spades"};
public String[] deck // initialize to size of 52 for the number of
cards there are in a standard deck (NO JOKERS)
```

These arrays will be important, and for that reason I want you to put the "ranks" and "suits" array in your code EXACTLY as you see it above. When we populate our deck we are going to store each card exactly as we say them, for example "Ace of Spades" or "3 of Clubs".

Class Methods:

```
class CardDeck
   -public String getIdentificationString()
   -public CardDeck()
   -public String getCard(int index)
   -public String getFirst()
```

```
    -public String getLast()
    -public String randomCard()
    -public String royalFlush()
```

public String getIdentificationString(), this will be found in almost all of your programs for the year this just returns a string with the program number and letter, your first and last name, and your cssc number. For example:

```
Program 2b, Firstname Lastname
```

public CardDeck(), this will be your constructor for your class and will be where you fill your "deck" array.
This one is going to be a bit tricky to populate but I know you guys can do it. I'm going to give you some hints to get started. For starters, use a for-loop within a for-loop, one for the ranks and one for the suits. Initialize the index tracker of the "deck" array OUTSIDE of these for-loops. Remember to store each card in "deck" as you say it: "rank" of "suit". I know this is going to be tricky but come by office hours of either the TAs or Professor Kraft we are going to be here to help you all.

public String getCard(int index), return the card at the given index.

public String getFirst(), return the first card in "deck".

public String getLast(), return the last card in "deck".

public String royalFlush(), return a String of cards that are contained in a Royal Flush poker hand. If you don't know what a Royal Flush consists of, you can give it a quick google search and you'll be able to figure it out. But the way I would like you to return it is all the cards on one line with a comma then a space in between each card.

Help

Be sure to get started with plenty of time to work on this. If you need any help with anything be sure to stop by Professor Kraft's office hours, or any of the TAs that are available in the lab. You can also put your questions in the discussion to see if any of your fellow classmates might be able to assist you. I'm sure all of us would be glad to help you out. Have fun and go learn something!

LAB ACTIVITY | 20.6.1: Program 2b: Card Deck 25 / 25 ✓

Submission Instructions

Compile command

```
javac CardDeck.java -Xlint:all -encoding utf-8
```

We will use this command to compile your code

Upload your files below by dragging and dropping into the area or choosing a file on your hard drive.

CardDeck.java Drag file here
or
Choose on hard drive.

Submit for grading

Latest submission - 8:21 PM on 09/16/19

Submission passed all tests ✓

Total score: 25 / 25

□ Only show failing tests

Download this submission

1: Unit test ^ 4 / 4

Test getIdentificationString()

2: Unit test ^ 4 / 4

Tests getCard(19)

3: Unit test ^ 4 / 4

Tests getFirst()

4: Unit test ^ 4 / 4

Tests getLast()

5: Unit test ^ 4 / 4

Tests randomCard()

Your output

```
You gave me: 8 of Spades
```

6: Unit test ^ 5 / 5

Tests royalFlush()

Your output

```
You gave me a good hand: 10 of Hearts, Jack
```