

CompSci 251

Assignment 1

Due Sep. 17th, 2018

Program Specification

This program will mimic a deck of cards. You will create a deck of cards, select how many people are playing the game, deal out 5 random cards to each player, then display each person's cards. After dealing out one round, you will ask the user if they wish to replay the round. You will finish writing a program I have started for you. I have provided a main method along with a template to follow.

Requirements

Download the Java file from D2L and load it into eclipse. I placed "//TODO" markers throughout the program, these are areas you need to complete. I have also added comments above each method describing how they should operate. Your program will load with errors, that is because some of the methods require a return type which you have to add.

Do not change any of the method names or parameters.

It is ok if your program crashes when reading input for how many people are playing. A crash could occur from using `scanner.nextInt()` and you input a letter by accident, or if you use `Integer.parseInt(String value)` and the String passed in is not a number in String format.

When complete, export your Java file to a zip folder and submit it to D2L.

Further Explanations

Your deck of cards is based on a boolean array of size 52. True means a card is still in the deck, false means it is not. The index numbers of the boolean array will be used to convert that number to its corresponding card. It will be up to you to determine how the conversion is done, but each index should map to a specific output.

For example, the number 4, which is index 4 of the array, could map to the String value "4D". Just make sure that the number 4 always maps to "4D".

Depending on how you do the conversion will determine what numbers to convert to what cards. I have provided two String arrays at the top of the assignment you may use in your conversions. They are visible to the entire program, but only should be used in `convertCard(int card)`.

The players are represented by a 2D array of Strings. The total number of players is user specified. Each row will represent one player, and the columns associated with that row will their cards. The total number of players allowed will be from 2 to 8. Below is an example how the array will work.

	0	1	2	3	4
0	"AC"	"3D"	"8S"	"2C"	"AH"
1	"9D"	"9H"	"7C"	"KH"	"10S"

Above we have two players. player[0]'s hand is represented by the row 0, and player[1]'s hand is represented by row 1. It will be your job to fill this array so each player has a full hand.

Below is a sample run.

How many people are playing? 2

Player 1 hand: 7D 2D 2C 5D 4D

Player 2 hand: 10H 8H 9H AS 2H

Would you like to deal again? (Type "no" to quit.) yes

How many people are playing? -1

Please re-enter a number between 2 and 8. How many people are playing? 5

Player 1 hand: 4C AH 8D AS 9C

Player 2 hand: 9H 8S JH 6S 6H

Player 3 hand: AD 3S 2D 2H 5D

Player 4 hand: 9D AC 7C KS JC

Player 5 hand: 7H 2C 10C 3H 10D

Would you like to deal again? (Type "no" to quit.) yes

How many people are playing? 999

Please re-enter a number between 2 and 8. How many people are playing? 8

Player 1 hand: QD 7H 5S JC 4S

Player 2 hand: 3H 2C 6S QH 7C

Player 3 hand: KH 3D JS 6C 10S

Player 4 hand: 2D AS 2S 5H 9S

Player 5 hand: JD AC 4C 8D JH

Player 6 hand: 10H QC 4D KC AH

Player 7 hand: 8S 10C 6D 9C 7S

Player 8 hand: 5D QS AD 5C KS

Would you like to deal again? (Type "no" to quit.) no

Thank you, goodbye