CPSC 1301, Computer Science I Programming Exercise 05

Week 05, Cards

1 Introduction

This activity consists of three programming exercises. The following exercises are open book and open note. You are free to use any written documentation you wish. However, these are individual exercises, and you cannot consult with each other in writing your programs. Name your program `Cards_lastname.py`.

This programming exercise has three parts. The grade for each requirement is indicated, for a maximum of 100 points. At a minimum, your program must compile successfully and run. This exercise uses Python lists. You are required to implement the functions using lists.

A starter template is shown below, but you do not have to use this. A sample output is also shown, and your exercise output must be similar to this output.

2 Exercise requirements

You are to implement three functions: `getNewDeck()`, `getShuffledDeck()`, and `getHands()`.

initial variables   You are given two variables, which are lists, as follows:

```
suits = ['Clubs','Diamonds','Hearts','Spades']
values = ['2','3','4','5','6','7','8','9','10','Jack','Queen','King','Ace']
```

getNewDeck(suit, values)   will take two list parameters (suits and values) and will return a list containing 52 items. Each item will be a "card" with a value and a suit, such as the `5 of Clubs` or the `Jack of Hearts`. The list will be a sorted list from the `2 of Clubs` to the `Ace of Spades`. Initialize the return list using two nested loops.

getShuffledDeck(newDeck)   will take one parameter (the new, sorted deck returned by the previous function) and return a shuffled deck. The shuffling must be randomized. See the discussion of `random.randint()` below. One way to do this is to iterate through the list, generating a random integer for each item, and swapping the current item for the random element. Another way would be to remove a random "card" from the new deck and append it to the shuffled deck, stopping when the new deck is empty. *Do not use any builtin methods — you must implement this by hand from scratch!*

getHands(shuffledDeck   will take one parameter (the shuffled deck returned by the previous function) and return a list of four lists named `north`, `east`, `south`, and `west`. This function will be responsible for taking the shuffled deck and dealing the deck into four hands of 13 cards each.

random.randint   Please check the documentation and make sure you know how to use this. Note that the starter template imports this method on line 7. The documentation states: " **random.randint(a, b)** Return a random integer N such that $a \leq N \leq b$."

3 Starter template

```python
1   #!python
2   # Name: Cards.py
3   # Author: Charles Carter
4   # Date; May 25, 2021
5   # Purpose: this is the Cards exercise, list operations
6
7   from random import randint
8
9   def hello():
10      print("Hello from 'Cards.py'")
11
12  def getNewDeck(suits, values):
13      """
14      Function: getNewDeck
15      Arguments: list of Suits suits, list of Values values
16      Returns: a new list for a new deck, nd
17      Purpose: creates a string like "X of Y" and appends the string to a new list by iterating
                through two lists using a nested for loop
18      """
19      #print( "called getNewDeck()")
20      pass
21      return nd
22
23  def getShuffledDeck(nd):
24      """
25      Function: getShuffledDeck
26      Arguments: list of a new ordered deck, nd
27      Returns: a new list for a shuffled deck, nd
28      Purpose: iterates through an ordered list randomizing (shuffling) the elements
29      """
30      #print( "called getShuffledDeck(newDeck)")
31      pass
32      return nd
33
34
35  def getHands(sd):
36      """
37      Function: getHands
38      Arguments: list of shuffled deck, sd
39      Returns: a new list of four lists, [N, E, S, W] for the hands
40      Purpose: creates four new lists representing hands, and returns a list of the hands
41      """
42      #print ('called getHands(shuffledDeck) returns ("north","east","south","west")')
43      pass
44      return [n, e, s, w]
45
46  #main function executes the defined functions
47  if __name__ == '__main__':
48      hello()
49      suits = ['Clubs','Diamonds','Hearts','Spades']
50      values = ['2','3','4','5','6','7','8','9','10','Jack','Queen','King','Ace']
51
52      newDeck = getNewDeck(suits, values)
53      print("New deck:", newDeck, "\n")
54
55      shuffledDeck = getShuffledDeck(newDeck)
56      print("Shuffled deck:", shuffledDeck, "\n")
57
58      [north, east, south, west] = getHands(shuffledDeck)
59      print("North:", north, "\n")
60      print("East:", east, "\n")
61      print("South:", south, "\n")
62      print("West:", west, "\n")
```

4 Sample output

```
Hello from 'Cards.py'
New deck: ['2 of Clubs', '3 of Clubs', '4 of Clubs', '5 of Clubs', '6 of Clubs', '7 of Clubs', '8 of Clubs',
'9 of Clubs', '10 of Clubs', 'Jack of Clubs', 'Queen of Clubs', 'King of Clubs', 'Ace of Clubs', '2 of Diamonds',
```

'3 of Diamonds', '4 of Diamonds', '5 of Diamonds', '6 of Diamonds', '7 of Diamonds', '8 of Diamonds',
'9 of Diamonds', '10 of Diamonds', 'Jack of Diamonds', 'Queen of Diamonds', 'King of Diamonds', 'Ace of Diamonds',
'2 of Hearts', '3 of Hearts', '4 of Hearts', '5 of Hearts', '6 of Hearts', '7 of Hearts', '8 of Hearts',
'9 of Hearts', '10 of Hearts', 'Jack of Hearts', 'Queen of Hearts', 'King of Hearts', 'Ace of Hearts',
'2 of Spades', '3 of Spades', '4 of Spades', '5 of Spades', '6 of Spades', '7 of Spades', '8 of Spades',
'9 of Spades', '10 of Spades', 'Jack of Spades', 'Queen of Spades', 'King of Spades', 'Ace of Spades']

Shuffled deck: ['9 of Hearts', 'Jack of Diamonds', '3 of Clubs', '5 of Diamonds', 'Queen of Clubs',
'7 of Spades', '8 of Diamonds', '2 of Clubs', '2 of Diamonds', '5 of Clubs', '7 of Diamonds',
'10 of Diamonds', 'King of Spades', 'Queen of Hearts', '9 of Clubs', '4 of Spades', 'Jack of Spades',
'Ace of Spades', 'Jack of Clubs', '8 of Hearts', '3 of Hearts', '8 of Clubs', '5 of Spades',
'6 of Hearts', 'King of Clubs', 'King of Diamonds', '3 of Spades', '4 of Hearts', 'Queen of Spades',
'8 of Spades', 'Ace of Hearts', '10 of Spades', 'King of Hearts', '4 of Clubs', '2 of Spades',
'6 of Clubs', '10 of Hearts', '6 of Spades', '6 of Diamonds', '9 of Diamonds', '10 of Clubs',
'Jack of Hearts', '2 of Hearts', 'Queen of Diamonds', '5 of Hearts', '7 of Hearts', 'Ace of Diamonds',
'Ace of Clubs', '4 of Diamonds', '9 of Spades', '3 of Diamonds', '7 of Clubs']

North: ['9 of Hearts', 'Queen of Clubs', '2 of Diamonds', 'King of Spades 'Jack of Spades', '3 of Hearts',
'King of Clubs', 'Queen of Spades', 'King of Hearts', '10 of Hearts', '10 of Clubs', '5 of Hearts', '4 of Diamonds']

East: ['Jack of Diamonds', '7 of Spades', '5 of Clubs', 'Queen of Hearts', 'Ace of Spades', '8 of Clubs',
'King of Diamonds', '8 of Spades', '4 of Clubs', '6 of Spades', 'Jack of Hearts', '7 of Hearts', '9 of Spades']

South: ['3 of Clubs', '8 of Diamonds', '7 of Diamonds', '9 of Clubs', 'Jack of Clubs', '5 of Spades',
'3 of Spades', 'Ace of Hearts', '2 of Spades', '6 of Diamonds', '2 of Hearts', 'Ace of Diamonds', '3 of Diamonds']

West: ['5 of Diamonds', '2 of Clubs', '10 of Diamonds', '4 of Spades', '8 of Hearts', '6 of Hearts',
'4 of Hearts', '10 of Spades', '6 of Clubs', '9 of Diamonds', 'Queen of Diamonds', 'Ace of Clubs', '7 of Clubs']