# BIS 420 PROGRAMMING FOR DATA SCIENCE

PRAJAKTA POHARE
CHAPTER 18 EXERCISE 18.3
ILLINOIS STATE UNIVERSITY

Write a Deck method called deal_hands that takes two parameters, the number of hands and the number of cards per hand, and that creates new Hand objects, deals the appropriate number of cards per hand, and returns a list of Hand objects.

```
class Card:
    suit_names = ['Clubs', 'Diamonds', 'Hearts', 'Spades']
    rank_names = [None, 'Ace', '2', '3', '4', '5', '6', '7',
                  '8', '9', '10', 'Jack', 'Queen', 'King']

    def __init__(self, suit=0, rank=2):
        self.suit = suit
        self.rank = rank

    def __str__(self):
        return f"{Card.rank_names[self.rank]} of {Card.suit_names[self.suit]}"

class Hand:
    def __init__(self, label=''):
        self.cards = []
        self.label = label

    def add_card(self, card):
        self.cards.append(card)
    def __str__(self):
        return f"{self.label} hand:\n" + '\n'.join(str(card) for card in self.cards)


import random


class Deck:
    def __init__(self):
        self.cards = [Card(suit, rank)
                      for suit in range(4)
                      for rank in range(1, 14)]

    def shuffle(self):
        random.shuffle(self.cards)

    def deal_hands(self, num_hands, cards_per_hand):
        hands = [Hand(f"Hand {i+1}") for i in range(num_hands)]
        for i in range(cards_per_hand):
            for hand in hands:
                if self.cards:
                    hand.add_card(self.cards.pop())
        return hands

    def __str__(self):
        return '\n'.join(str(card) for card in self.cards)

deck = Deck()
deck.shuffle()
```

```
hands = deck.deal_hands(4, 5)


for hand in hands:
    print(hand)
    print()
```

```python
class Card:
    suit_names = ['Clubs', 'Diamonds', 'Hearts', 'Spades']
    rank_names = [None, 'Ace', '2', '3', '4', '5', '6', '7',
                  '8', '9', '10', 'Jack', 'Queen', 'King']

    def __init__(self, suit=0, rank=2):
        self.suit = suit
        self.rank = rank

    def __str__(self):
        return f"{Card.rank_names[self.rank]} of {Card.suit_names[self.suit]}"

class Hand:
    def __init__(self, label=''):
        self.cards = []
        self.label = label

    def add_card(self, card):
        self.cards.append(card)

    def __str__(self):
        return f"{self.label} hand:\n" + '\n'.join(str(card) for card in self.cards)

import random

class Deck:
    def __init__(self):
        self.cards = [Card(suit, rank)
                      for suit in range(4)
                      for rank in range(1, 14)]

    def shuffle(self):
        random.shuffle(self.cards)

    def deal_hands(self, num_hands, cards_per_hand):
        hands = [Hand(f"Hand {i+1}") for i in range(num_hands)]
        for i in range(cards_per_hand):
            for hand in hands:
                if self.cards:
                    hand.add_card(self.cards.pop())
        return hands

    def __str__(self):
        return '\n'.join(str(card) for card in self.cards)

deck = Deck()
deck.shuffle()

hands = deck.deal_hands(4, 5)

for hand in hands:
    print(hand)
    print()
```