```python
# TheDeckClass.py
""" Contains a class that can be used to manipulate a
deck of
playing cards.
"""

from random import shuffle as shuffle
from TheCardClass
import *

class Deck(object):
    """ Represents a deck of playing cards.


    Attributes:
        DeckOfCards: list of Card objects
                  n: int


    n is the number of cards in the deck.

        The "top" of the deck is
self.DeckOfCards[0]
        The "bottom" of the deck is self.DeckOfCards[self.n]

"""

    def __init__(self):
        """ Returns a reference
to a Deck object that represents a traditional
        deck of 52 playing cards.

"""
        self.n = 52
        self.DeckOfCards = []
        for suit in
range(4):
            for rank  in range(1,14):
                card = Card(suit,rank)

        self.DeckOfCards.append(card)

    def __str__(self):
        """
Returns a string s such that print s
        nicely displays self,
        """

      s = []
        for card in self.DeckOfCards:
            s.append(str(card))

return '\n'.join(s)

    def pop_card(self,Where=None):
        """ Returns
a Card from self and removes that Card from
        self.DeckOfCards
        PreC: self has at
least one card. Where is a string
        that determines how the card is extracted:

    Where == 'Top'     the card at the top of  the deck
              Where == 'Bot'     the card
at the bottom of the deck
              Where == None     the card is randomly selected

"""
        if Where=='Top':
            c = self.DeckOfCards.pop(0)

elif Where=='Bot':
            c = self.DeckOfCards.pop()
        elif Where==None:
```

```python
        k = randi(0,self.n-1)
        c = self.DeckOfCards.pop(k)
        self.n -= 1

        return c

    def add_card(self,c):
        """ Adds the Card c to self.

        PreC: c is a Card
        """
        self.DeckOfCards.append(c)

        self.n += 1

    def shuffle(self):
        """Randomly permutes the
entries in self.DeckOfCards
        """
        shuffle(self.DeckOfCards)


    def sort(self):
        """ Permutes the entries in self.DeckOfCards so
that they
        are sorted with respect to the __cmp__ function in the class Card.

"""
        self.DeckOfCards.sort()
```