

King of Hearts, lincerta from <https://pixabay.com/en/hand-playing-cards-paper-table-1492256/>, CC0 Public Domain

Free Response Question 3

Your goal is to simulate a standard deck of cards so that the cards can be shuffled and used in various card games. For this activity, the class is StdCardDeck.

A standard deck of cards contains four suits ( hearts, spades, diamonds, and clubs). Each suit has 13 numbers from 2 through Ace.

The StdCardDeck has two arrays of String[]. The first array is for cardSuit for the four suits of cards. The second array is for cardValue for the 13 values (2 through Ace). There is also a Card array called cards which should be instantiated in the constructor.

Pre-Condition: (Assume these arrays are pre-defined)

     String[] cardSuit = {“heart”, “spade”, “diamond”, “club”};

     String[] cardValue = {“2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “10”, “Jack”, “Queen”, “King”, “Ace”};

The Card class has the following Card Constructor:

      public Card(String suit, String value)  
      {  
           // implementation not shown  
      }

The StdCardDeck class has the following instance variable:

     Card[] cards;

Instructions: You may print and hand write your answers to the following questions. This will be how you will be completing this part on the actual AP Exam. To submit if you hand write, you will need to take an image or scan and submit. For this practice, you may also enter your answers directly into this document, save and submit

(a) Write a constructor that creates and stores the 52 Cards that StdCardDeck needs in order to function. The result of this constructor should be a deck where each Card is unique and has both a suit and a value.

/\*\*Creates a deck of cards with 52 unique Card objects, each

\* with a suit and a value.

\* Precondition: StdCardDeck must have two arrays: one for

\* cardSuit and one for values, both String[] arrays

\*/

public StdCardDeck()

{

int index = 0;

for (int count = 0; count < cardSuit.length() - 1; count++)

{

for (int count2 = 0; count2 < cardValue.length() - 1; count2++)

{

Card newCard = new Card(cardSuit[count], cardValue[count2]);

cards[index] = newCard;

index++;

}

}

}

(b) Write a method called cardSwap, which takes an integer for the index of the first Card and an integer for the index of the second Card, and then swaps the two cards in-place in the Card[] array.

/\*\*Given two indices, the method moves the corresponding Cards

\* in the Card[] array to the opposite indices

\*/

public void cardSwap(int firstIndex, int secondIndex)

{

Card tempCard = Cards[secondIndex];

Cards[secondIndex] = Cards[firstIndex];

Cards[firstIndex] = tempCard;

}

(c) Assume that the method from Part (b) functions properly and the Random class has been imported. Create a method called shuffle to take the created StdCardDeck and perform an in-place shuffle of the cards for use in a card game. Since this is meant to be used in a card game, the cardSwap function must create a different order of Cards every time.

/\*\*Performs an in-place swap of every card with a random second card to swap their values\*/

public void shuffle( )

{

Random rand = new Random();

for(int count = 0; count < Cards.length(); count++)

{

int random1 = rand.nextInt(53);

int random2 = rand.nextInt(53);

this.cardSwap(random1, random2);

}

}