In this programming project you will be designing classes to mimic a deck of playing cards. For those of you who do not play cards regularly, you should know that there are typically 52 cards in a standard deck composed of the following 13 ranks:

Ace

Two

Three

Four

Five

Six

Seven

Eight

Nine

Ten

Jack

Queen

King

Furthermore, these cards consist of one of the following four suits:

Clubs

Diamonds

Hearts

Spades

For our purposes, we will assume the ranking of the suits above (taken from the game of bridge) - clubs will be the lowest rank, followed by diamonds, hearts and then spades.

Your job is to write code for a class called Deck (you may need other classes for your implementation as well). The header for the Deck class and its methods are shown on the next page.

**Grading**

* You MUST use an ArrayList (not an Array) as the underlying structure for the Deck class.
* You must document your code thoroughly for this project.
* You must also write a main routine that tests your code. It should demonstrate all the features listed on the next page.
* Although the user will not see it, you should create a class called Card to hold each card.
* You must submit a hardcopy (on paper) of your code and output. Email will not be accepted.

**public class Deck {**

**public Deck()**

// default constructor creates an empty deck (with no cards in it)

**public Deck (int n)**

// constructor creates a deck with n cards chosen randomly with no duplicates.

// Therefore Deck(52) would create a full deck.

// Deck(0) would be equivalent to Deck() (i.e. would create an empty deck)

// Calls to this constructor with values greater than 52 (or less than zero)

// should create a full deck (52 cards) and should also result in a warning message being

// displayed on the console.

**public int findCard(String rank, String suit)**

// findCard() should return the index of the location of the card given in the parameters.

// If the card is not located in the deck, it should return a value of -1.

**public void sortDeck()**

//sortDeck() should sort the deck. Sorting of a deck should place all the clubs at the lowest

// indicies, followed by diamonds, then hearts, then spades. Within each suit, the cards should

// be in rank order. Attempts to sort an empty deck should result in no action or error.

**public void shuffleDeck()**

//shuffleDeck() should arrange the cards in the deck randomly.

**public String toString()**

// Overloaded toString() function prints the deck. If the deck d1 contained the Ace of Spades,

// Two of Hearts, and Five of Clubs (in that order), here is what would be printed by calling

// System.out.println(d1); [AS, 2H, 5C]

**public boolean addCard(String cardRank, String cardSuit)**

// adds the given card to the deck. Returns true if the addition was successful.

// returns false if an attempt is made to add a duplicate card (i.e. the card is already in the deck)

// or if parameters do not specify a valid card (i.e. d1.addcard(“Eleven”, “Spades”); )

**public boolean removeCard(String cardRank, String cardSuit)**

// removes the given card from the deck. Returns true if the card was found and removed.

// cards to the left should keep their existing indicies, cards to the right should move one down.

**public boolean isFull()**

// returns true if the deck has 52 cards in it

**public boolean isSorted()**

//returns true of the deck is currently sorted

} // end class Deck