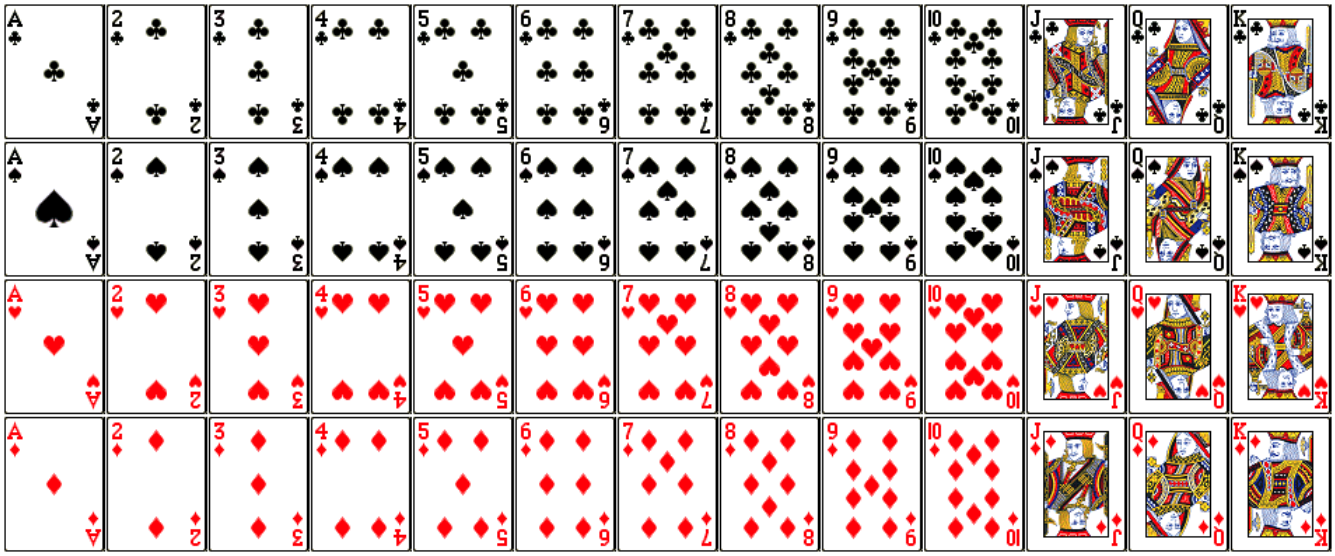
Model 1 Deck of Cards

There are 52 cards in a standard deck. Each card has one of ***13 ranks*** (1=Ace, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11=Jack, 12=Queen, and 13=King) and one of ***4 suits*** (0=Clubs, 3=Spades, 2=Hearts, and 1=Diamonds). For example, `new Card(12, 2)` would construct the Queen of Hearts.

The following deck is represented by an array of `Card` objects. The array is one-dimensional, but the cards are shown in four rows (because of the paper margins).



Questions (25 min) Start time:

1. What is the index (in the array above) of the following cards?

a) Ace of Clubs

b) Jack of Clubs

c) 2 of Spades

d) Queen of Spades

e) 7 of Hearts

f) King of Diamonds

2. Write the following statements using one line of code each.

a) Declare and initialize a `Card` array named `deck` that can hold 52 cards.

b) Construct the Ace of Clubs, and assign it as the first element in `deck`.

c) Construct the King of Diamonds, and assign it as the last element in `deck`.

3. Describe how you could repeat code from the previous question to construct the entire deck of cards (without having to type 52 statements).

4. Discuss the following code as a team:

```
int index = 0;
int[] suits = {0, 3, 2, 1};
for (int suit : suits) {
    for (int rank = 1; rank <= 13; rank++) {
        deck[index] = new Card(rank, suit);
        index++;
    }
}
```

a) What is the overall purpose of the code?

b) Why is the `suits` array not just {0, 1, 2, 3}? (See Model 1.)

c) Why does the code use an enhanced for loop for `suit`?

d) Why does the code use a standard for loop for `rank`?

e) What is the purpose of the `index` variable?

5. Write a method named `inDeck` that takes a `Card[]` representing a deck of cards and a `Card` object representing a single card, and that returns `true` if the card is somewhere in the deck.

6. Describe what the following code does and how it works. (Note: You've come a long way this semester, to be able to understand this example!)

```
public static Card[] sort(Card[] deck) {
    if (deck == null) {
        System.err.println("Missing deck!");
        return null;
    }
    Card[] sorted = new Card[deck.length];
    for (Card card : deck) {
        int index = card.position();  // returns suit * 13 + rank - 1
        sorted[index] = card;
    }
    return sorted;
}
```

a) What is the overall purpose of the code?

b) What is the purpose of the if statement?

c) Does this method modify the `deck` array? Justify your answer.

d) How does the `sort` method know where to put each card?

7. Identify the following Java language features in the previous question.

a) variables

b) decisions

c) loops

d) methods

e) arrays

f) objects