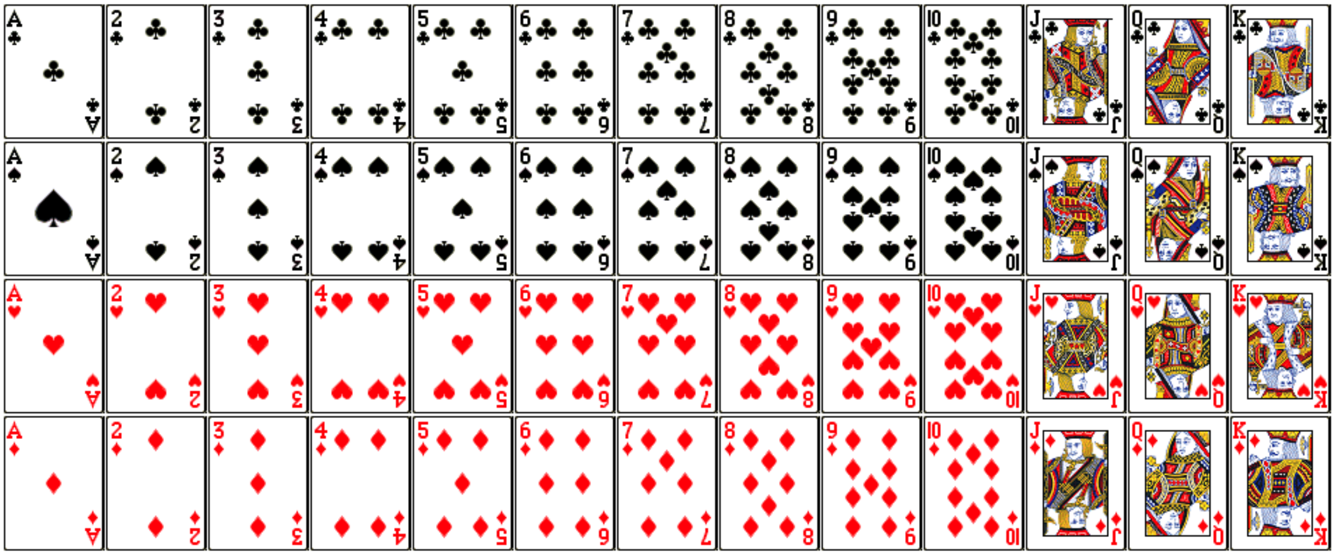
Model 1 Deck of Cards

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 0

b) Jack of Clubs 10

c) 2 of Spades 14

d) Queen of Spades 24

e) 7 of Hearts 32

f) King of Diamonds 51

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.

```
Card[] deck = new Card[52];
```

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

```
deck[0] = new Card(1, 0);
```

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

```
deck[51] = new Card(13, 1);    // or deck[deck.length - 1]
```

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

Use nested for loops to iterate each possible `rank` and `suit`, construct that card, and assign it to the next index in the `deck` array.

4. Discuss the following code as a team:

```java
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?

It creates a deck of cards in the order shown in Model 1.

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

Because the picture shows the suits in a different order.

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

The code iterates the suits out of order, as specified in the array.

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

The code iterates the ranks in order; no array is needed for that.

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

To keep track of where to store the next card (not based on rank and suit).

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.

```java
public static boolean inDeck(Card[] deck, Card card) {
    for (Card c : deck) {
        if (c != null && c.equals(card)) {
            return true;
        }
    }
    return false;
}
```

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!)

```java
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?

This example sorts an array of cards.

b) What is the purpose of the if statement?

It avoids `NullPointerException` if `deck` is invalid.

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

No; it creates and returns a new array named `sorted`.

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

It computes the `position` based on its rank and suit.

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

a) variables deck, sorted, card, index

b) decisions if (deck == null)

c) loops for (Card card : deck)

d) methods sort, println, position

e) arrays deck, sorted

f) objects "Missing deck!", card