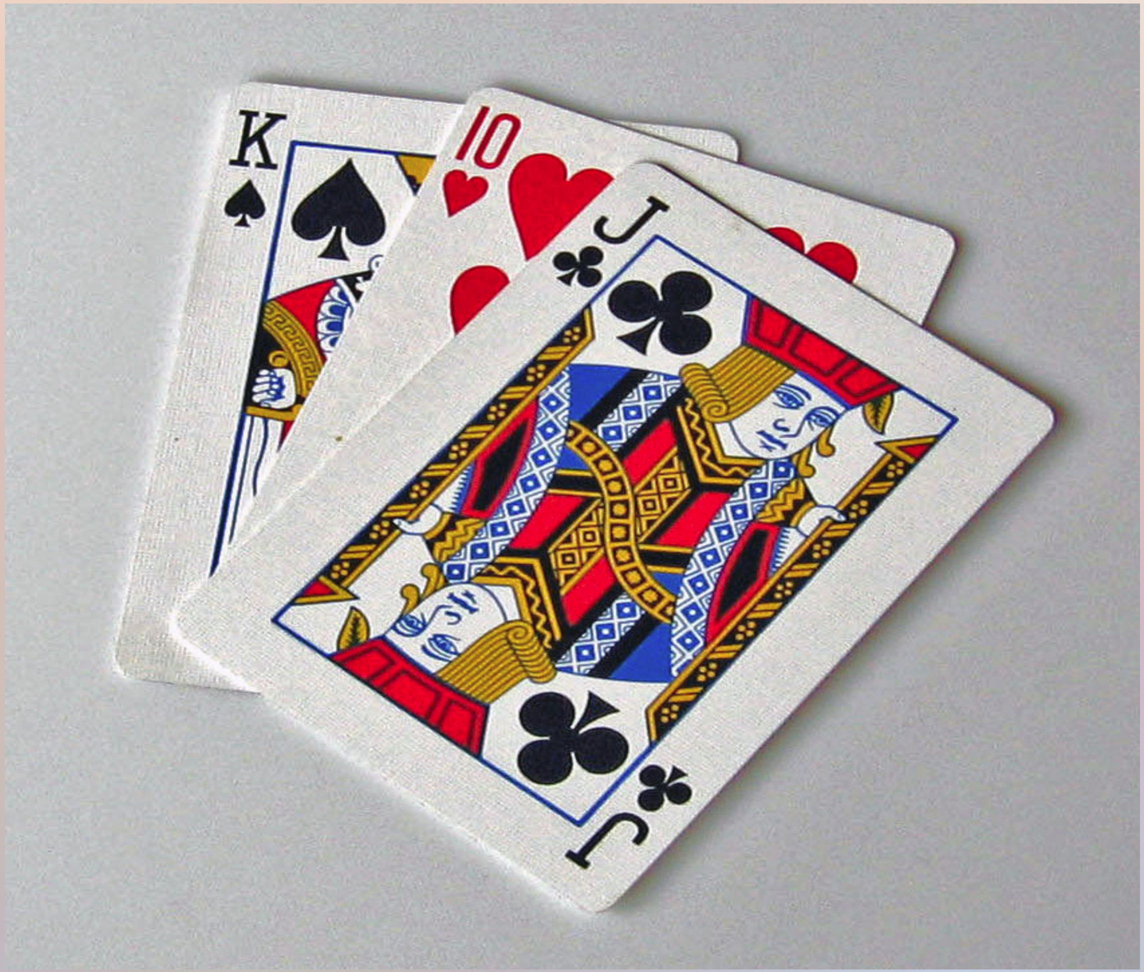
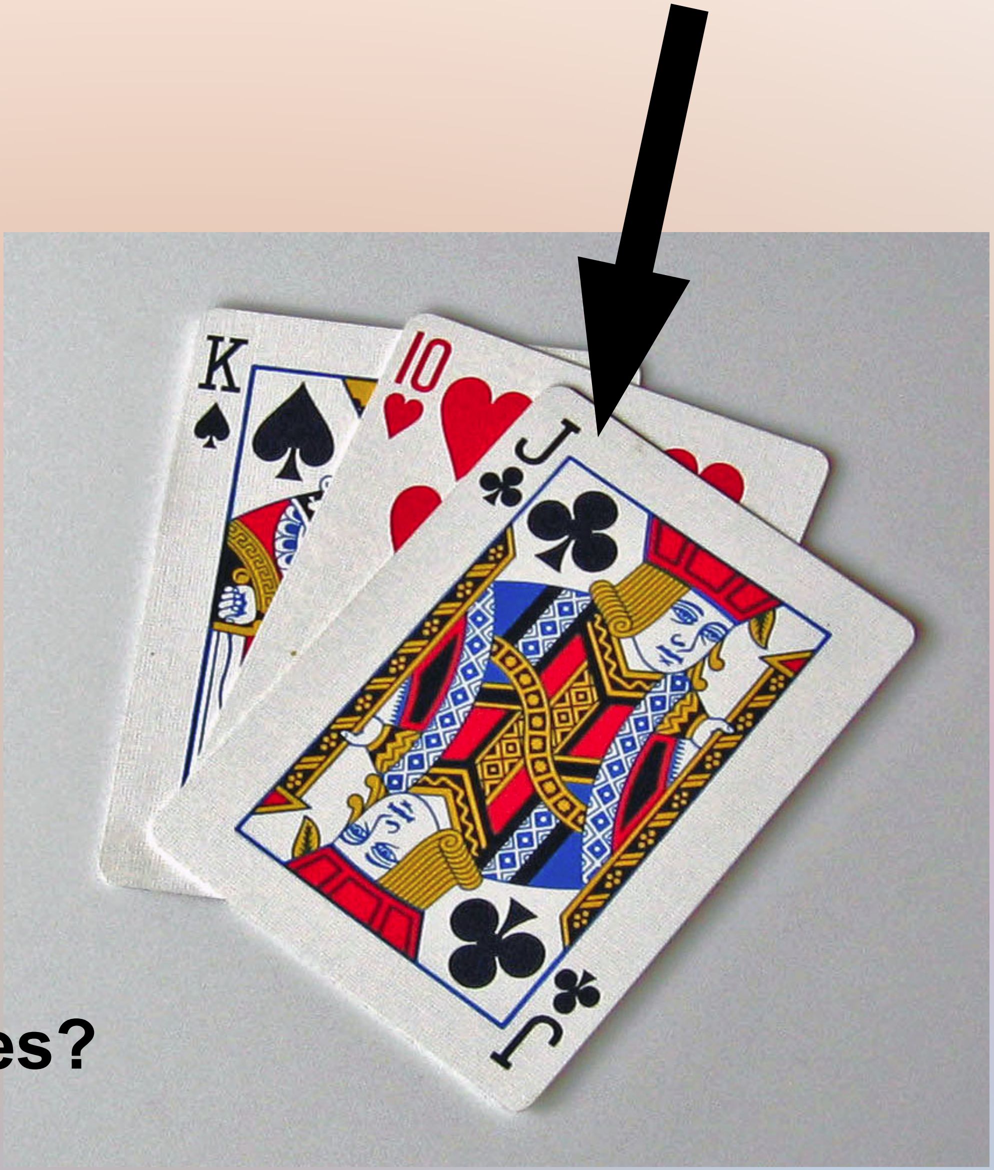
Documenting a Program for presentation

First explain the problem to be solved and the model used

Problem solving:
What is a deck of cards?
How can cards be modeled?



This card can be card 18 in the deck and is a Jack of Clubs



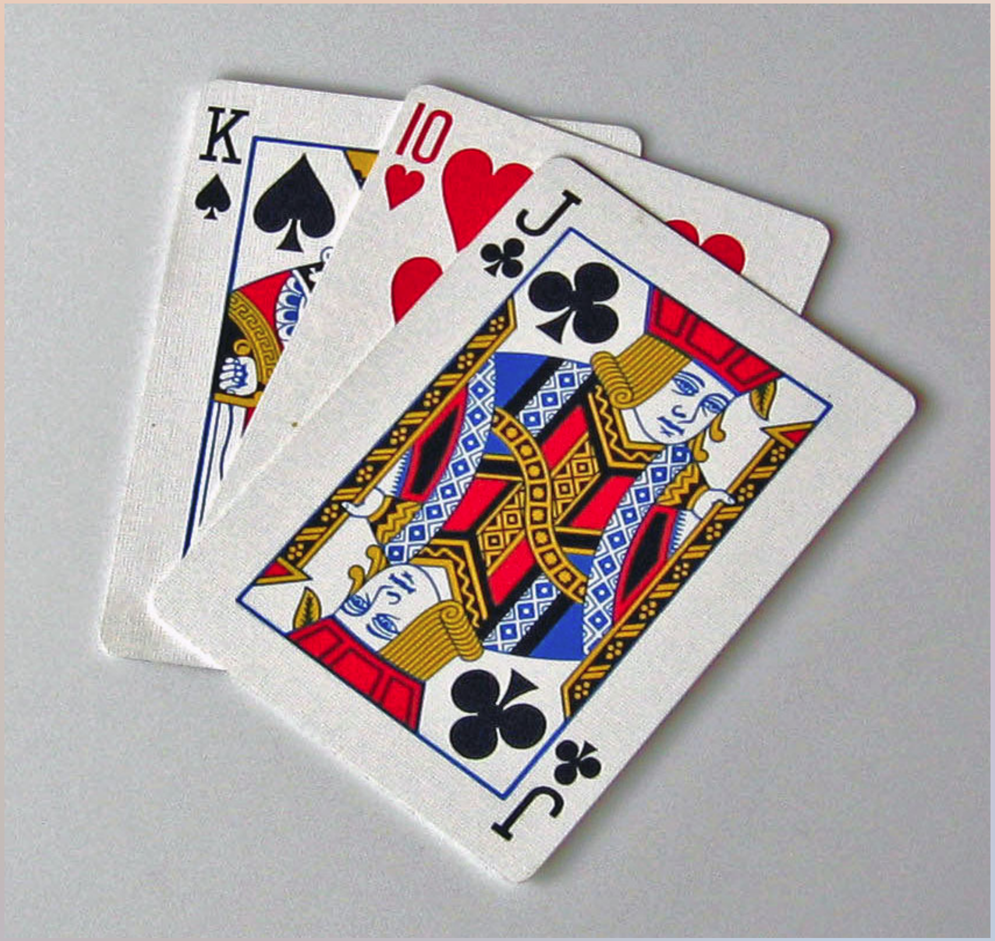
The card number is 18
Clubs is the suit
Jack is the face value

Problem solving:
How many cards?
How many suits?
How many face values?

Problem Solving: Specification

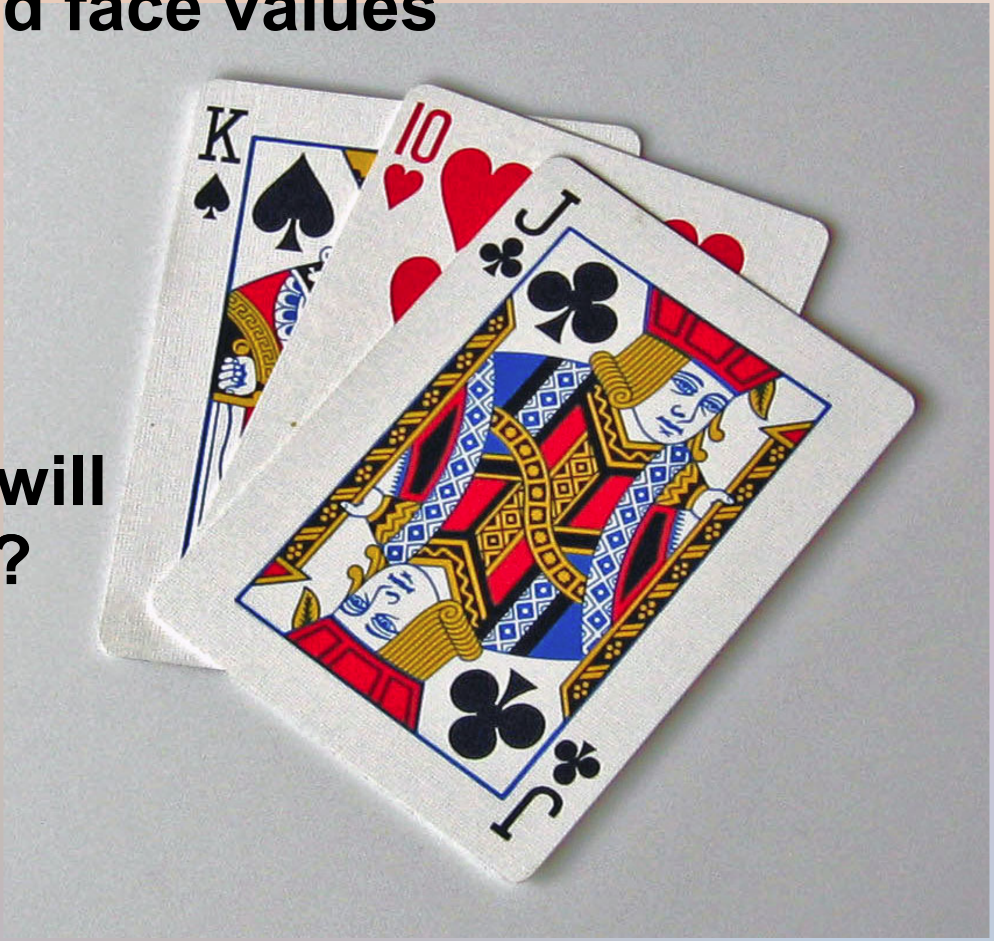
Face Values are
1-Ace
2-Two
3-Three
4-Four
5-Five
6-Six
7-Seven
8-Eight
9-Nine
10-Ten
11-Jack
12-Queen
13-King

Suit values are
0-Spades , 1-Diamonds
2- Clubs, 3-Hearts



So the card deck array will need
an array row for each card

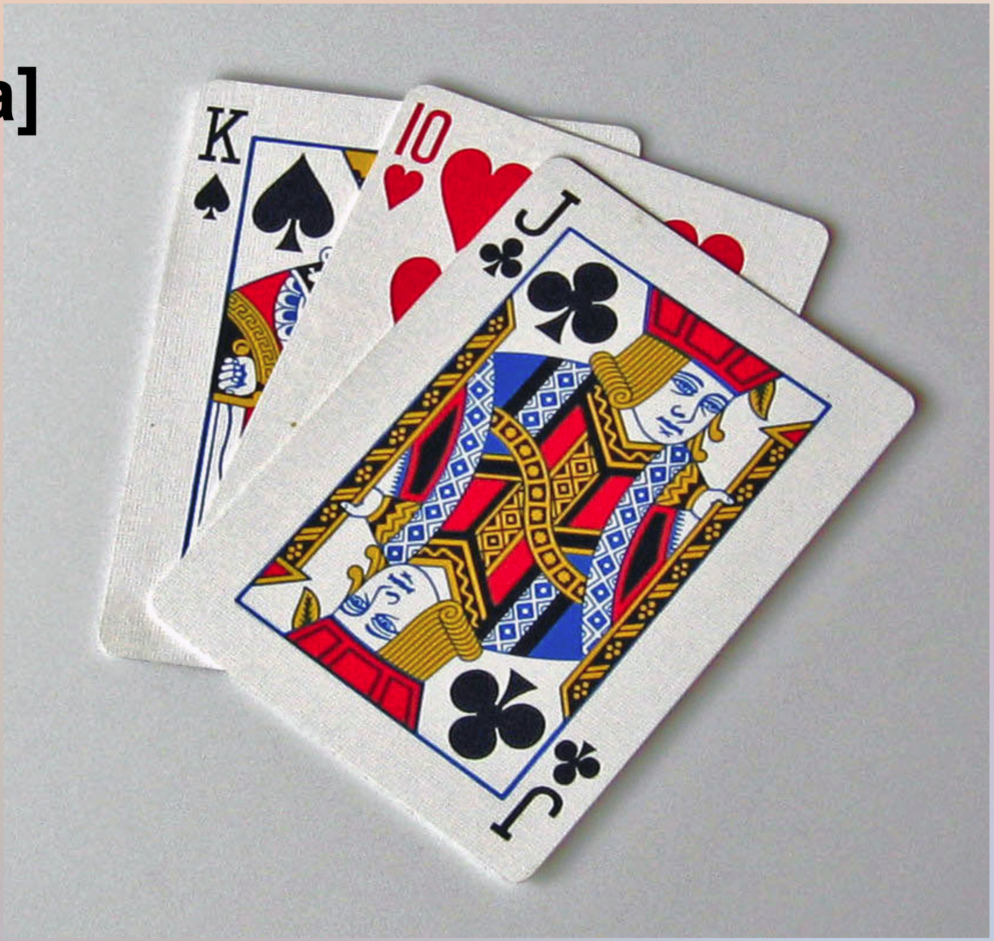
and each card row has an index number plus will need
two elements for suit and face values



Problem Solving:
How many dimensions will
CardDeck array require?

How much information needed for each card?
How many dimensions does the array need?

CardDeck[Row][Data]



Problem solving:

Each card has an index number for the card row

and elements for the card suit and face values

How many elements needed for each card?

Problem Solving:

The *first dimension*: each element represents
card row in deck

The *second dimension* represents
the two card characteristic elements

first element holds suit number of card

second element holds face number of card

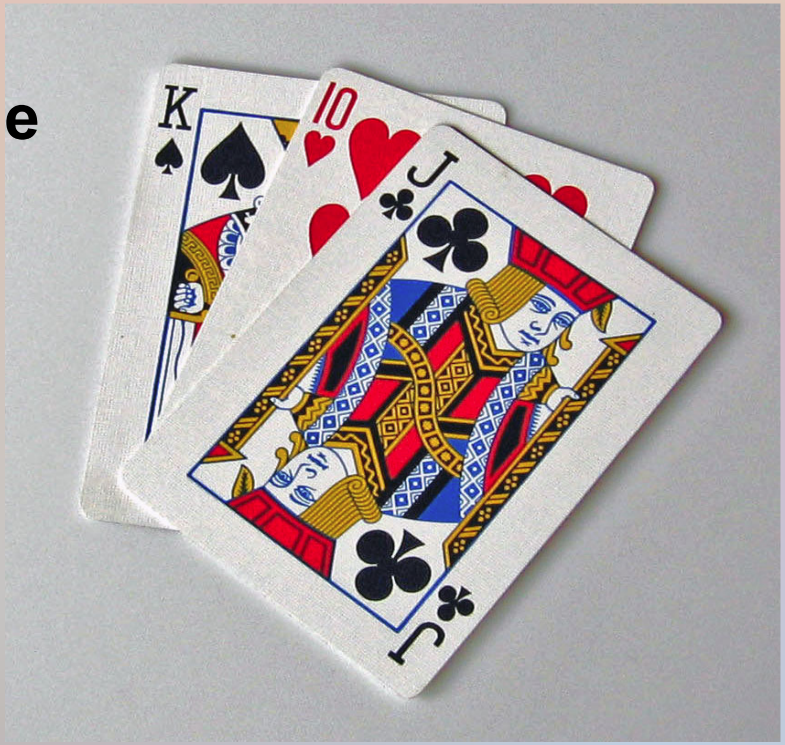
According to prior definition table

suit

Card[18][0] = 2 (Clubs)
Card[18][1] = 11(Jack)

face

Card Row Number



Card#	Suit #	Face Value
Row#	Column [0]	Column [1]
0	0 - Spades	0 - Ace
1	0 - Spades	1 - Two
2	0 - Spades	2 - Three
3	0 - Spades	3- Four
4	0 - Spades	4 - Five
5	0 - Spades	5 - Six
6	0 - Spades	6 - Seven
7	0 - Spades	7 - Eight
8	0 - Spades	8 - Nine
9	0 - Spades	9 - Ten
10	0 - Spades	10 - Jack
11	0 - Spades	11 - Queen
12	0 - Spades	12 - King
13	1 - Diamonds	1 - Ace
14	1 - Diamonds	2 - Two
15	1 - Diamonds	3 - Three
16	1 - Diamonds	4- Four

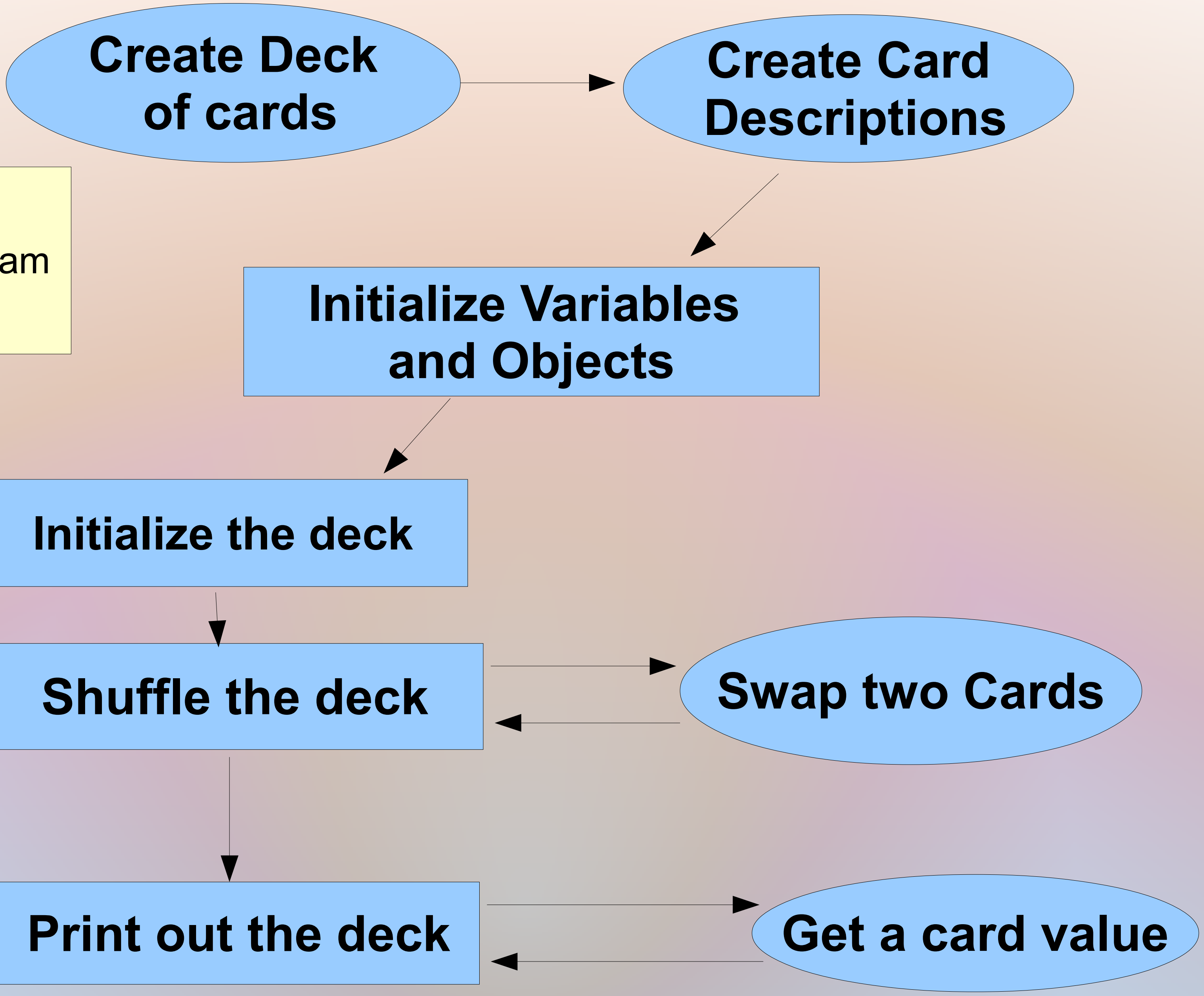
**A multi-dimension array
can be visualized as
a table with
Rows being the first dimension
and Columns being the second**

What does the program do?:

Creates and use an array model for a deck of cards.

The deck of cards is created and initialized
Then deck of cards is shuffled and printed.
A print routine outputs a single card value

Process Diagram



Program Hierarchy

```
Main | Srand
     |
     | InitDeck
     |
     | ShuffleDeck  | SwapCards
     |
     | PrintCard    | GetPlayValue
```

Function Name: InitDeck

Initialize the deck with card values
void InitDeck(int deck[NCARDS][NPROPS]);
Calling Arguments: card deck
Return Argument: none

```
Sequence:
    Create local loop increment variable
    Loop through suits
        Loop through faces
            Set the suit value
            Set the face value
    The loops should initialize 52 cards total
```

Function Name: Shuffle

Shuffle the card deck

```
void ShuffleDeck(int deck[NCARDS][NPROPS]);
```

```
Sequence:
    Create local variables ( src, dest)
    Loop through each dest card row (all 52 cards)
        create a random source card number
            call function to swap the src and dest
```

Function Name: SwapCards

```
Swap two cards in deck
void SwapCards(int deck[NCARDS][NPROPS], int src, int dest)

    Create 'temp' local variable

    Do once for suit and again for face values
      fill temp with dest suit: temp = deck[dest][0];
      fill dest with src suit:   deck[dest][0] = deck[src][0];
      fill src with temp suit:  deck[src][0] = temp;
```

Function Name: PrintCard

```
Print a card suit and face value
void PrintCard(int deck[NCARDS][NPROPS], int card)

    Create local variables: suitvalue, facevalue, playval
    fill suitvalue & facevalue from card in deck
        suitvalue = deck[card][0];
    get the play value of card
        playvalue = GetPlayValue(deck, card);

    print string value of the cards –
        card value = index of string so,
    printf( “%s of %s \n”,face[facevalue],........
```

Function Name: GetPlayValue

```
Determine the play value of a card
int GetPlayValue(int deck[NCARDS][NPROPS], int card)
    Create local variables: facevalue, playvalue
    fill facevalue with of card row face value
    determine play value of card
        if(facevalue <=10) then return facevalue
        else
        return 10 ; Jack, Queen, King
```