

# Interview Card Game

This exercise has been created in an attempt to gauge the level of a candidate's ability to think about solving problems, and actually implement a solution. Full access to any internet resources is allowable as are all tools provided by the IDE.

## ***The Problem***

This is a very simple card game. There is a single deck of 52 standard poker cards. A group of players is each dealt a single card. The winner of the game is then decided by which player has the highest card. The rank of cards is determined first by the face values of the card: 2,3,4,5,6,7,8,9,10,J,Q,K, and A, and then by the suit, Heart, Diamond, Club and Spade in that order. Examples, an Ace will always beat all other face cards, and if two players both have an Ace, then the Ace of Hearts will beat all other Aces in the deck.

## ***The Program***

Shuffler.java is the main class for this program and will "play" the game with the following steps:

1. Create the Players
2. Create the Deck
3. Shuffle the Deck
4. Deal a card to each player
5. Determine the winner of the game

The card and deck classes have been created, but the constructor and the shuffle methods of the Deck class are missing implementations.

After the deck is shuffled, each player is dealt a card. The winner of the game is the player with the highest card. The Shuffler.java class has a determineWinner() method that is missing it's implemented.

The program has three TODO tags located in the three main areas for implementation, but the solution can modify any part of the existing code that that candidate decides on.

When completed, the program should produce the following output.

```
2H 3H 4H 5H 6H 7H 8H 9H 10H JH QH KH AH 2D 3D 4D 5D 6D 7D 8D 9D 10D JD QD KD AD 2C 3C 4C 5C 6C 7C
8C 9C 10C JC QC KC AC 2S 3S 4S 5S 6S 7S 8S 9S 10S JS QS KS AS
QC 10C 10H 8S 5D 10D 3H 5S 2H 3C 4H AC 9D 2C 4S QS AS 8D 2S KC KD 8C JH 9C 4C 7D 9S JS 7C 6C 8H 5C
4D 6S AH KH 2D AD 9H 10S JC 3D KS QH 6H QD 7H JD 3S 5H 6D 7S
Joe 10C
Bill 10H
Jim QC
The winner is Jim QC
```

## ***Discussion***

After completion of this exercise, the candidate should be able to discuss the program from an architecture viewpoint. The candidate should present reasoning behind the solution that they created.

The candidate should also be able to discuss and whiteboard the program in terms of basic UML diagrams, especially class and sequence diagrams.

Finally the candidate should be able to discuss ideas about what would be required to implement this program in different ways, like as a web page or as a client application.