

# 1 Exercise

For this lab you may work together but each person will submit their own programs. As usual, there is no file sharing allowed.

This exercise is a simplified version of the child's card game of War. In War each child has a shuffled deck of playing cards, a standard poker deck. They each turn over their first card, the highest face value wins (with Ace being low, then 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K). The winner of the round takes both cards. If the face values are the same then this is "war" where each player turns over the next three cards. The highest face value wins all the cards. In the event that after the three cards are turned over the face values are again the same three more cards are turned over, and so on until there is a winner of the round. At the end of the game the child with the most cards wins the game. Clearly there is no strategy to this game and the winner is completely determined by the shuffle of the cards. Hence it is a nice game for children but not for older persons.

Our game is going to be a simplified version of this. The intent is to get experience using and updating class structures (i.e. Objects).

Here is our game. Each of two players has a shuffled deck of playing cards, a standard poker deck. They each turn over their first card, the highest face value wins (with Ace being low). The winner of the round gets one point. If the face values are the same then no one wins that round. This goes on for the 52 cards and the person with the highest score wins the game.

Create a program that will simulate this game using the Card and Deck classes we discussed in lecture. Note that you will not need any user input for this. Have the program print out the shuffled decks (which may help with debugging) and then each round of the game with a running score. At the end have the program print the winner of the game or if the scores are equal have it print that the game was a draw. Note that you may need to update the Card class since the method that returns the worth of a card equates the 10, J, Q, and K where in this game they have different worths.

## Decks

```

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

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

## Game

```

7S  2H  Score: 1  0
7D  6H  Score: 2  0
3S  QH  Score: 2  1
JH  10H Score: 3  1
3C  QS  Score: 3  2
5S  JC  Score: 3  3
AH  3C  Score: 3  4
8D  6D  Score: 4  4
7H  5C  Score: 5  4
KD  6C  Score: 6  4
10D 10S Score: 6  4
KH  KD  Score: 6  4
AD  8C  Score: 6  5
10H 3S  Score: 7  5
```

```
2D  QC  Score:  7   6
5C  AS  Score:  8   6
2H  2C  Score:  8   6
4H  JD  Score:  8   7
QS  4S  Score:  9   7
JS  JS  Score:  9   7
9H  2D  Score: 10   7
6C  AD  Score: 11   7
QC  5S  Score: 12   7
6D  7S  Score: 12   8
8S  10C  Score: 12   9
2C  4C  Score: 12  10
9S  KH  Score: 12  11
KC  KS  Score: 12  11
5D  9S  Score: 12  12
6S  AC  Score: 13  12
KS  4D  Score: 14  12
10S 4H  Score: 15  12
4C  7H  Score: 15  13
2S  3H  Score: 15  14
6H  2S  Score: 16  14
3D  JH  Score: 16  15
4S  7D  Score: 16  16
QH  5D  Score: 17  16
8H  3D  Score: 18  16
7C  QD  Score: 18  17
5H  6S  Score: 18  18
3H  AH  Score: 19  18
9C  KC  Score: 19  19
AC  9C  Score: 19  20
AS  8D  Score: 19  21
JD  9H  Score: 20  21
QD  10D  Score: 21  21
8C  7C  Score: 22  21
JC  8H  Score: 23  21
4D  9D  Score: 23  22
10C 8S  Score: 24  22
9D  5H  Score: 25  22
```

Player 1 won the game.

## 2 Submit

1. The Java code files for main program and the files for the Card and Deck classes.
2. A Word, LibreOffice, or text file containing at least three runs of the program.