Design and implement the following program using object oriented approach in C++. This program simulates a simple card game that can be played between 2 or more players [4].

A game consists of a number of rounds between **P** players. At the start of each round, every player receives **C (= 52/P)** cards. The remaining **52%P** cards are turned face up so that everyone can see them. A single game consists of **C** rounds in which each player plays a single card in turn. The player of the first card in the first round of a game is decided randomly. Thereafter, the winner of each round plays the first card of the next round of that game, if there is one. After the first card of a round has been played, the remaining players play one card each in turn. These players must follow the suit [2] of the first player if possible. A round is won by the player who plays the highest card of the lead suit. A game is won by the player or players who have won the most rounds in that game.

The purpose of the assignment is to program an implementation of the Player class that is capable of taking a legal part in a game. The interface between the game's manager and each player is defined in terms of the member functions of the player class. The interactions at the start of a new game enable a Player object to know what cards it holds (`receiveCard`), how many other players there are in the game, and which cards are spare (i.e. held by no-one - `notifyOfSpare`). In each round, a player is informed of whether it must lead the round (`leadRound`), whether it must follow a particular suit, and which other cards have been played in the current round. Most interactions take place between players and manager by passing around pointers to `Card` objects. A Player does not need to know how such an object is implemented but can enquire of it its `suit` and `value`.

Players must follow suit, whenever possible. If they have no card of the same suit, they may play any other card in which case that particular player will not be considered in the round.

Guidance on Design

The game should be run by a manager who deals the cards and keeps players informed of which cards are left over after the deal, which cards have been played, and so on.

The following considerations are not part of the assignment, but you might like to bear them in mind when thinking through the design.

- How easy would it be to add the notion of a trump suit to the design?
- Could one or more of the players be a human player or cooperation between the program and a human advisor?
- Could the manager take part in the game?

You should first prepare identification of the Classes that are needed, their responsibilities (public interfaces), and their Collaborators (relationship between objects/classes). You may do this using any existing techniques you might have come across during this course.

The implementation of this game should use console for (nice, clean and well-formatted [5]) output and keyboard for input. However, saving and loading the game's current state in the file is a required feature that must be implemented in a way that an unfinished game can be loaded/stored to/from the file. Once the state of the game is loaded from the file, the game should continue as usual. You can use simple file streams for this purpose.

Assessment will be based on followings:

- Design aspects
- Implementation aspects
- Tactical aspects

Deadline for submission is last week of the semester. You will have to use the concepts of OOP (composition, aggregation, association, inheritance, polymorphism etc.) for better design and efficient implementation during this assignment.

Useful links

1. http://www.daniweb.com/software-development/cpp/threads/278453/card-game-hw-assignment-hell
2. http://nifty.stanford.edu/2004/EstellCardGame/assignments/cardgamepart1.html
3. http://www.cs.kent.ac.uk/people/staff/djb/t204/cards.html
4. http://nifty.stanford.edu/2004/EstellCardGame/
5. http://www.cprogramming.com/tutorial/iomanip.html