Video Poker game (Jacks or Better)

Objectives:

- Use scoped enumerations.
- Use classes and objects.
- Create a program menu.
- Write code that is "const-correct."
- Use exceptions

You are to write a game of "Five Card Video Poker (see: http://en.wikipedia.org/wiki/Video_poker)

Guidelines:

- due date: TBD
- to be submitted to blackboard
- Menus will work with uppercase and lowercase letters.
- You must follow the structure of the program as shown below.
- The code must meet the class style guide
- The names of variables and functions should be descriptive.

Instructions:

- You must create VideoPoker class to play “Jacks or Better” rules video poker.
  - it must have a play() method that plays one round of poker
  - it must have a showPayTable() method that displays the pay table
  - it must have a welcome() method that prints a welcome message, including a brief introduction to the game.
  - it will have a Deck object and a Hand object as private members that are used to play the game
- your main() program will only use the VideoPoker object, the VideoPoker object will contain and manage the other classes to play the game.
- Only your VideoPoker object will use cin/cout.
- You must create a VideoPokerHand class that represents a poker hand, and the rules of the game. This class contains the ability to rank hands and calculate pay out amounts.
  - The hand class will use a scoped enum **PokerHand** of the hand types (royal flush, full house, etc).
  - The hand class will have a static map paySheet that maps PokerHand to payout amount for poker hands
  - The hand class will have a static map handNames that maps PokerHand to the name of the hand as a string
  - The hand class will have a toString() method that will return the name of the current hand as a string (using handName map)

  - the Hand class must have a score() method that returns the type PokeHand defined by your scoped enum
  - your Hand class **should** have the following state (private members)
    - std::vector< Card::CardPtr> _hand; // your hand
    - std::map< int, int > _faceCount; // number of each face
    - std::vector<bool> _drawCard; // draw or hold flag
  - you must use std algorithms to identify the hand types
  - You should implement the following methods to use in scoring hands
    - bool isRoyalFlush() const;
    - bool isStraightFlush() const;
    - bool isFourOfAKind() const;
    - bool isFullHouse() const;
    - bool isFlush() const;
    - bool isStraight() const;
    - bool isThreeOfAKind() const;
    - bool isTwoPair() const;
    - bool isJacksOrBetter() const;
    - bool isPair() const;
- You may use the Card and Deck classes from previous homework or those provided in the starter project.
- Your Deck class may use pointers to Card objects, as in the starter project, or solid card objects. If you use pointers you **must** use smart pointers.
- You must create "Util.cpp" files and "Util.h" to contain utility functions such as showMenu(). The file that contains your main() function should have only the main() function definition in it. Util.h should have most of the “view” in it.
- Problems in Card, Deck, and Hand that can not be handled locally should cause a meaningful exception to be thrown.

The definitions of different hands

- **High Card:** The face and suit of the highest card in the hand
- **Pair**: Formed by two cards of the same rank (face value)
- **Jacks or Better**: a pair of jacks or higher ranked cards (J, Q, K, A)
- **Two Pair:** Formed by two pairs
- **Three of a Kind:** Formed by three cards of the same rank (face value)
- **Straight:** Composed of five cards of consecutive rank (face value)
- **Flush**: Composed of five cards of the same suit
- **Four of a Kind**: Formed by four cards of one rank accompanied by any card.
- **Straight Flush**: Composed of five cards of consecutive ranks of the same suit
- **Royal Flush**: a Straight Flush from 10 to Ace.

A hand value is always the highest possible value for that hand. For example even though a **three of a kind** hand contains a **pair** it is not ranked as a **pair** it is ranked as a **three of a kind.**

Pay Sheet:
from Wikipedia:

Hand	1 credit
Royal Flush	250
Straight Flush	50
Four of a kind	25
Full House	9
Flush	6
Straight	4
Three of a kind	3
Two Pair	2
Jacks or Better	1
Theoretical Return	98.4%

example app:
There is an example implementation of video poker application on Blackboard. You may play it to get an example of what your game might look like.