Final Project

Object Oriented Design
Due 23:59 on April 28
(Group project, two students in a group)

Implement a computer Casino program that has three games: Slot machine, Blackjack, and Texas hold 'em. For blackjack game, the computer program is the dealer and provides interface for user to play. The following rules should be implemented and the game should follow the dealing routine as defined.

Slot machine rules:

1. Bet 1, 2, 5, 10 dollars
2. Three reels
3. 10 symbols each reel – you can define your symbols
4. A symbol can occupy several stops on the multiple reel
5. Random Number Generator for Selecting the Reel Stop Positions
6. Three rows of symbols are shown at any given time
7. Win: Three symbols in the center row are the same – 300 times the bet
8. Win: Two symbols in the center row are the same – 30 times the bet
9. Win: Three symbols on the screen are the same – 5 times the bet
10. Win: Two symbols on the screen are the same – 2 times the bet

Your program should implement all the rules, provide interface for users to play, randomly generate the three rows of symbols, show them on screen, compute if a win occurs, collect or pay the bet.

Blackjack rules:

1. User's initial balance -- $1000
2. Minimum bet – $5 and maximum bet -- $100
3. Multiple players – as many as 5
4. Win – closer to 21 than the dealer (computer) without going over 21
5. Player's cards are all face up (shown to all players and the computer)
6. Dealer's first card is face down, the rest are face up
7. An Ace can count as either 1 or 11
8. The cards from 2 through 9 are valued at their face value
9. The 10, Jack, Queen, and King are all valued at 10
10. The value of a hand is simply the sum of the point counts of each card in the hand, For example, a hand containing (5,7,9) has the value of 21
11. The dealer must continue to take cards ("hit") until his total is 17 or greater
12. Blackjack rule: A blackjack, or natural, is a total of 21 in your first two cards. A winning blackjack pays the player odds of 3 to 2. A player blackjack beats any dealer total other than blackjack, including a dealer's three or more card 21. If both a player and the dealer have blackjack, the hand is a tie or push.
13. The player has the option to either draw another card to the hand ("hit"), or stop at the current total ("stand").

14. When the player has two cards, the player can “double down". Doubling down allows the player to double his/her bet and receive one, and only one, additional card to the hand.
15. When the player has two cards, the player can “surrender”, which offers the choice to fold the player’s hand, at the cost of half of the original bet. The player must make that decision prior to taking any other action on the hand
16. When the player is dealt a matching pair of cards, you have the ability to split the hand into two separate hands, and play them independently. The rule allows a player to split up to 3 times, making 4 separate hands, with 4 separate bets

Routine:

1. Place bet
2. Once all the bets are made, the dealer will deal the cards to the players. He'll make two passes around the table starting at his left (your right) so that the players and the dealer have two cards each
3. The dealer will flip one of his cards over, exposing its value
4. The players cards will be dealt face-up
5. Once the cards are dealt play proceeds around the table, starting at the first seat to the dealer's left, also called first base. Each player in turn indicates to the dealer how he wishes to play the hand
6. After each player has finished his hand, the dealer will complete his hand, and then pay or collect the player bets.

Your program should implement all the rules, provide interface for users to play, shuffle cards, deal cards, compute who wins, take or pay the bet.

Texas hold 'em rules and routine is at http://en.wikipedia.org/wiki/Texas_Hold%27em and http://www.briggsoft.com/docs/pmavens/PMHoldem.htm . Your program shall support up to 8 players. You can follow the example at http://www.thepokerpractice.com/ .

If you decide to form a group of three students, you have to implement one more game: **Casino Craps**. Your program shall follow the rule of play in Wikipedia at http://en.wikipedia.org/wiki/Craps . The computer will roll the dice. Your game starts with one player and allow new players to join. Players can leave at the end of each bet. You can follow the example at http://www.crapsage.com/free_craps.php .

Grading:

- No compilation error – 0 otherwise
- Blackjack – 30%
- Slot machine – 30%
- Poker – 40%

- For groups with three students, you will need to implement the Casino Craps. The grading distribution is
  - Blackjack – 20%
  - Slot machine – 20%
  - Poker – 20%
  - Casino Craps – 30%

Your implementation should has satisfy the following requirements (the percentage indicating how much you will lose if you do not meet the requirement):

- Everything should be implemented with classes other than the main interface and the definitions of the classes are separated from their implementations – 10%
- Use operator overloading – 5%
- Use inheritance – 5%
- Use polymorphism – 5%
- Use template – 5%
- All classes have constructor and destructor – 5%
- Use vector – 5%
- Use const and static – 5%
- Use new and delete – 5%
- Report – 50%. Your report should indicate where all the required components are implemented. It also should include an instruction and an example input for testing.

Extra Credit 1 (10 %) : Implement a graphical user interface for all games.

Extra Credit 2 (10 %) : Implement two computer players in poker with artificial intelligence algorithm. Explain the algorithm and how it is implemented in your program.