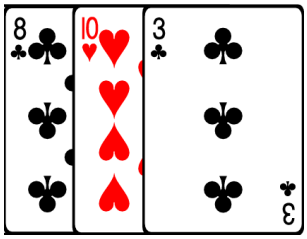
CSCI 3110 (Barbosa S18)
Project 4: Stack the Deck

Due: **Wed Feb 28 by 11:59 PM** – may be turned in until Mar 4 by 11:59 PM with reduced points (per Project guidance found in the course syllabus).

Assignment ID: **proj4**
File(s) to be submitted: **proj4.cpp, card.h, card.cpp, deck.h, deck.cpp, player.h, and player.cpp.**

Above files must be uploaded individually.
Submit only above files. DO NOT submit a zip file.



Objectives: 1) Use of static variables; 2) Using multiple classes; 3) Operator overloading; 4) Overloading ostream; 5) Using friend functions.

Project description:
Write a C++ program to simulate a simple card game between two players. The game should proceed as follows: The 52 cards in a deck of cards are shuffled and each player draws three cards from the top of the deck. The remaining cards are placed in a pile face-down between the two players. Players then select one of the three cards in hand and simultaneously place the chosen card face-up on the game table. The player who places the highest ranking card on the table collects both cards and adds them to their pile (score). If both cards have the same value the hand is a draw and no points are accumulated. Following the completion of a hand, each player draws the top card from the deck to add to their hand. Play continues until all cards have been played. The winner is the player with the most points at game's end.

You will use a standard deck of 52 cards where there are thirteen cards from each of four suits: hearts, spades, diamonds, and clubs. The thirteen cards in point order are the 2-10 numbered cards, the face cards (Jack, Queen, King), and the Ace card. Points are distributed as follows: Ace=15, face cards=10, all other cards count as their numeric value.

Requirements:

1. Your program must be split into 7 files. There will be 3 classes (each with separate interface and implementation files), and a driver file. The requirements for these are specified below:

a) The *Card* class – This class represents an individual card

- Files must be named ***card.h*** and ***card.cpp***
- Class must be named ***Card***
- The interface (header file) is provided.
    i. You should implement the interface file in a .cpp implementation file
    ii. All data members must be of the type specified in the header file
    iii. All member functions in the interface file must be implemented as declared – However you have flexibility in how you choose to implement each

b) The *Deck* class – This is represents the deck of cards

- Files must be named ***deck.h*** and ***deck.cpp***
- Class must be named ***Deck***
- The interface (header file) is provided.
    i. You should implement the interface file in a .cpp implementation file
    ii. All data members must be of the type specified in the header file

iii. All member functions in the interface file must be implemented as declared – However you have flexibility in how you choose to implement each

c) The *Player* class – This class will represent the human and computer player – Play is autonomous

- Files must be named ***player.h*** and ***player.cpp***
- Class must be named ***Player***
- The interface (header file) is provided.
  i. You should implement the interface file in a .cpp implementation file
  ii. All data members must be of the type specified in the header file
  iii. All member functions in the interface file must be implemented as declared – However you have flexibility in how you choose to implement each

c) A driver, or client, file

- Must be named ***proj4.cpp***
- Must contain the line ***srand(1000);*** as the first line of the main function
- Must instantiate the card deck and the players, and control the game play, printing out the hand, scores, outcome (winner, loser, draw), and final winner and score as shown in the sample output.
- Output for this program must clearly and neatly show that the program works and that it works correctly. Your output should:
  1. Display the entire deck of cards at the beginning of the game, and after it has been shuffled
  2. For each hand, show the cards and score of each player before the play, the card each player plays, and the cards and score of each player after the play (but before the next card is drawn)
  3. At the end show the winner of the game and the winning score.
- Below is a sample output of a game's start and some late game hands as the game ends

2. Sample output:

a) Display of deck both before and after shuffling, and play of first couple of hands. Individual cards are output as card, suit, value. Example: 8 ♠ [8] is the 8 of spades which is worth 8 points; K ♥ [10] is the King of hearts with value 10. **Note**: depending on how you implement deck shuffling, the cards played may, or may not, match those below.

```
The original deck has the following cards
K♦[10]  Q♦[10]  J♦[10]  10♦[10]  9♦[9]  8♦[8]  7♦[7]
6♦[6]  5♦[5]  4♦[4]  3♦[3]  2♦[2]  A♦[15]  K♠[10]  Q♠[10]  J♠[10]
10♠[10]  9♠[9]  8♠[8]  7♠[7]  6♠[6]  5♠[5]  4♠[4]  3♠[3]  2♠[2]
A♠[15]  K♥[10]  Q♥[10]  J♥[10]  10♥[10]  9♥[9]  8♥[8]  7♥[7]  6♥[6]
5♥[5]  4♥[4]  3♥[3]  2♥[2]  A♥[15]  K♣[10]  Q♣[10]  J♣[10]  10♣[10]
9♣[9]  8♣[8]  7♣[7]  6♣[6]  5♣[5]  4♣[4]  3♣[3]  2♣[2]  A♣[15]

The deck after shuffling (top of deck listed first)
4♥[4]  J♥[10]  6♠[6]  7♠[7]  6♣[6]  K♥[10]  5♠[5]
5♣[5]  9♦[9]  10♣[10]  6♥[6]  Q♠[10]  Q♣[10]  K♣[10]  8♦[8]  8♣[8]
9♠[9]  7♣[7]  6♦[6]  A♠[15]  10♥[10]  2♦[2]  5♥[5]  Q♦[10]  4♣[4]
A♦[15]  2♠[2]  9♣[9]  3♣[3]  2♥[2]  9♥[9]  5♦[5]  J♦[10]  7♥[7]
4♦[4]  J♠[10]  3♦[3]  7♦[7]  3♠[3]  A♣[15]  10♠[10]  Q♥[10]  K♦[10]
10♦[10]  8♠[8]  K♠[10]  J♣[10]  4♠[4]  3♥[3]  2♣[2]  A♥[15]  8♥[8]

************** Hand 1 **************
Player 1's hand: 4♥[4] 6♠[6] 6♣[6] with score 0
Player 2's hand: J♥[10] 7♠[7] K♥[10] with score 0

Player 1 played card: 6♣[6] --- Player 2 played card: J♥[10]

Player 2 wins this hand
Player 1's hand: 4♥[4] 6♠[6] _____ with score 0
Player 2's hand: _____ 7♠[7] K♥[10] with score 16

************** Hand 2 **************
Player 1's hand: 4♥[4] 6♠[6] 5♠[5] with score 0
Player 2's hand: 5♣[5] 7♠[7] K♥[10] with score 16

Player 1 played card: 6♠[6] --- Player 2 played card: 7♠[7]

Player 2 wins this hand
Player 1's hand: 4♥[4] _____ 5♠[5] with score 0
Player 2's hand: 5♣[5] _____ K♥[10] with score 29
```

b) Last few hands of the game. **Note**: The deck is exhausted after hand 23. Play continues until payers are out of cards.

```
*************** Hand 23 ***************
Player 1's hand: 10♥[10] J♣[10] 3♥[3] with score 174
Player 2's hand: 4♠[4] 2♣[2] K♠[10] with score 160

Player 1 played card: 10♥[10] --- Player 2 played card: 4♠[4]

Player 1 wins this hand
Player 1's hand: _____ J♣[10] 3♥[3] with score 188
Player 2's hand: _____ 2♣[2] K♠[10] with score 160

*************** Hand 24 ***************
Player 1's hand: A♥[15] J♣[10] 3♥[3] with score 188
Player 2's hand: 8♥[8] 2♣[2] K♠[10] with score 160

Player 1 played card: A♥[15] --- Player 2 played card: K♠[10]

Player 2 wins this hand
Player 1's hand: _____ J♣[10] 3♥[3] with score 188
Player 2's hand: 8♥[8] 2♣[2] _____ with score 185

*************** Hand 25 ***************
Player 1's hand: _____ J♣[10] 3♥[3] with score 188
Player 2's hand: 8♥[8] 2♣[2] _____ with score 185

Player 1 played card: 3♥[3] --- Player 2 played card: 2♣[2]

Player 1 wins this hand
Player 1's hand: _____ J♣[10] _____ with score 193
Player 2's hand: 8♥[8] _____ _____ with score 185

*************** Hand 26 ***************
Player 1's hand: _____ J♣[10] _____ with score 193
Player 2's hand: 8♥[8] _____ _____ with score 185

Player 1 played card: J♣[10] --- Player 2 played card: 8♥[8]

Player 1 wins this hand
Player 1's hand: _____ _____ _____ with score 211
Player 2's hand: _____ _____ _____ with score 185

Player 1 wins with a score of 211
```

Note: You can use the following statements to print suit symbols:

```
cout << ((char)0x03); //print Heart symbol
cout << ((char)0x04); //print Diamond symbol
cout << ((char)0x05); //print Club symbol
cout << ((char)0x06); //print Spade symbol
```

3. Test your program - Use different initializations.

4. Code comments - Add the following comments to your code:

- A section at the top of the source file(s) with the following identifying information:

  Your Name
  CSCI 3110-00X (your section #)
  Project #X
  Due: mm/dd/yy

- Below your name add comments in each file that give an overview of the program or class.
- Place a one or two line comment above each function that summarizes the workings of the function.