CSCI 2170 Bonus programming assignment (50 pts)
(Due: 23:59, Sunday March 18th)

Write a C++ program which will play a card game called "Hearts". Hearts is a trick taking game in which the objective is to avoid winning tricks containing hearts; the queen of spades is even more to be avoided.

The program plays "Hearts" with four players: computer 1, computer 2, computer 3, and the user. The winner of the game is the person who has the ***lowest number of points*** when the game is over. Points in the game are calculated as follows: each card of suit HEART has points: those with face value less than 10 is worth 5 points, and the rest of the HEART cards worth 10 points. In addition, the Queen of Spades worth 100 points, the Club of Jack worth -100 points.

Each player is dealt 13 cards at the beginning of the game. The person who holds the 2 of CLUB must lead it to the first trick (2 of CLUB should be played on the first round). The suit of the first card played on each hand is called the **leading suit** for that hand. Then, the remaining three players must each play a card of the same suit if they have one. If not, they can play a card from any other suit. For each hand, the player with the highest valued card of the **leading suit** must collect the points. Then, this player must begin the next round in a similar fashion with a new leading suit. Therefore, 13 rounds will be played. The game is over after all 13 rounds have been played. In each round, the players should play in clockwise rotation. For example, if the user starts the round, the 2nd player is computer 1, and then computer 2, and computer 3. If computer 2 starts the round, the 2nd player is computer 3, followed by user, and then computer 1.

Your program simulates this game. At each round, your program must:

- Show the user's cards in a nice format such that the user can view all his or her remaining cards easily,
- Display the cards played by each player,
- Show points accumulated for each player, and
- Indicate who will lead the next hand.

To start the first card of a round,

- if it is the user's turn, the user may select any card
- if it is computer player's turn, the player selects a random card from his hand.

To play a card after other users have already started the hand, (aka to follow up on a card):

- if it is the user's turn, he can select an appropriate card of the required/leading suit from his remaining cards.
- If it is the computer players' turn, he randomly selects a card of the leading suit. IF there is no card of the leading suit left in hand, a random card is selected from his hand.

Program Development:

Download type.h, and the header file and implementation file for the CardClass and PlayerClass from the course page. The PlayerClass has been modified with added functions.

Complete the client program by adding code to play the entire "hearts" game. In your client program, define an array of PlayerClass objects as:

```
PlayerClass player[4];
```

player[0] is the user, player[1], player[2], and player[3] are the three computer players.

The header file for the modified PlayerClass is shown next:

```
class PlayerClass
{
    public:
      // default constructor
      // post-condition: count is assigned 0, score is assigned 0
      PlayerClass();

      // add one card to the player's hand
      // pre-condition: player has less than 13 cards, a card is supplied as input parameter
      // post-condition: player has one more card in hand, count is increased by 1
      void       AddCard(CardStruct);

      // prints out the current cards in the player's hand
      void       DisplayCards();

      // select to play the first card that has the suit supplied from the client program
      // if no card can be found that has the suit supplied by the client program, the first card
      // from the hand is played
      // pre-condition: there are >= 1 cards in hand
      // post-condition: a card is returned/played, count is decremented by 1
      CardStruct  FollowOneCard(suitType s);

      // Plays the first card of a round. A randomly card in the deck is selected/returned
      // pre-condition: there are >= 1 card(s) in the player's hand
      // post-condition: one card is played/returned, count is decremented by 1
      CardStruct  StartOneHand();

      // Checks to see if the player should lead the
      // first round in the game, e.g., check whether the player has 2 of club
      // pre-condition: the hand is full (have 13 cards)
      // post-condition: if this player has 2 of club, true is returned, otherwise, return false.
      bool                IsFirstLead();

      // The current score of the player is returned
      int        GetScore();

      // The points from the current round are added to the current player's score
      // pre-condition: the score from the current round is sent as input parameter
      //post-condition:  the  player's  score  is  increased  by  the  points  sent
      //             in through the parameter
      void       AddScore(int);

      // Plays the card selected by user. After the cards in the player is displayed to the user, the
      //  user selects a card by entering the number corresponding to the card.
      // pre-condition:  the card number selected by the user is supplied
      // post-condition: the card corresponding to the user choice is played/returned.
      //                    the number of cards in player's hand is decremented by 1
      CardStruct  PlaySelectedCard(int choice);

      // Return the number of cards the player has
```

```
    // post-condition:   the number of cards is returned
    int       GetCount();

    // Checks to see if the card the user chooses is a valid choice. If the user has cards of the
    // leading suit, the card he chooses to play has to match the leading suit.
    // pre-condition:  the player's choice of card number and the current leading suit
    //                 are sent as input parameters
    // post-condition: returns true if
    //       (1) user has cards of leading suit and the choice card is of that suit
    //       (2) user does not have card of leading suit
    //      and returns false otherwise
    bool      IsValidChoice(suitType, int);

  private:
    CardStruct   hand[MAX_PLAYER_CARDS];
    int          count;   // keeps record of the number of cards the player has in hand
    int          score;   // keeps score for the player
};
```

Instruction to turn in the program:

Soft copy:

- login the ranger system with www.cs.mtsu.edu/nx,
- login to PeerSpace through the web browser provided by the ranger system, click on *tools|Assignments* to submit your softcopy.

Hard copy:

- Create a script file by following the steps below: First, navigate to the directory where your program source file is located, then follow the steps below:

```
ranger% script bonus
ranger% pr –n –t –e4 hearts.cc
ranger% aCC  CardClass.cpp PlayerClass.cpp hearts.cc
ranger% a.out                ← play one game to completion
ranger% exit
ranger% lph bonus
```

- Enclose the hardcopy of the program and the program evaluation sheet in a folder