LAB #3 – Practice the Big Three

In order to get credit for the lab, you need to be checked off by the end of lab. You can earn a maximum of 3 points for lab work completed outside of lab time, but you must finish the lab before the next lab. For extenuating circumstances, contact your lab TAs and Jennifer Parham-Mocello.

Reminder: All of our labs involve paired programming. You do not have to keep the same partner for each lab, but you MUST work with someone in each lab!!! First, find a partner for this lab. It can be the same partner from the previous lab or a different partner.

(7 pts) hand class
You will make sure you can write the constructors and the Big Three for the game and hand classes from assignment #2. Note, you may need the Big Three in more classes, but it depends if the class has dynamic memory.

First, work on the hand class, since you already have the card class (or struct) written!!!

class hand {
   private:
      card *cards;
      int num_cards;
   public:
};

• Write the default constructor, mutators, and accessors you need for the hand class. Here are some prototypes to get you started. After you complete these, make sure it you can set and get cards.

   hand(); //set cards to NULL and num_cards to zero
   void set_card(const int, const card); //the card to set and card
   card * get_cards() const; //return address of array of cards
   void create_cards(const int); //set num_cards and make card array
   int get_num_cards() const; //return number of cards in hand

• Now, write the “Big Three” for the hand class. Here are the prototypes to get you started.

   hand(const hand &); //copy constructor
   ~hand(); //destructor
   void operator=(const hand &); //assignment operator overload

Convince your TA that the constructors, mutators, accessors, and “Big Three” are working correctly for the hand class!!!

(3 pts) game class

Now, continue working on your assignment #2, and begin writing the game class.
First, make decisions on the constructors, mutators, and accessors you need for the hand class!!! Also, begin implementing a void play_game() function, which you will use to convince yourself that the deck is constructed correctly and you can get and set the number of players.

NOTE: Right now, comment out the players and game_dealer members, since you might not have the player or dealer classes written yet.

```cpp
class game {
    private:
        deck cards;
        player *players;
        dealer game_dealer;
        int num_players;
    public:
    }
```

For the remaining part of the lab, begin writing the player class focusing on the hand in the class!!! What are the constructors, mutators, and accessors you will need to interact with the player’s hand from the play_game() in the game class? For example:

```cpp
players[0].get_hand().create_cards(2)
players[0].get_hand().set_card(0, cards.remove_top_card())
players[0].get_hand().set_card(1, cards.remove_top_card())
```

```cpp
class player {
    private:
        hand p_hand;
    public:
    }
```

If you have enough time, implement the “Big Three” for the game class!!!

**Remember**, you and your partner will not receive lab credit if you do not get checked off before leaving each lab. Once you have a zero on a lab, then it cannot be changed because we have no way of knowing if you were there or not!!!