

## Assignment #2 – Begin Classes/OOP in C++

### Due: Sunday, 04/24/16, 11:59pm

**Grading:** EVERY assignment in this course is graded by demoing your work for 10 minutes with a TA. You are required to meet with a TA within two weeks from the due date to demo, and you receive an automatic 50 point deduction for failure to do so. If you miss a scheduled appointment, you will be penalized 10 points for rescheduling within 1 day (24 hours), 25 points within 7 days (1 week), and 50 points for anything outside of a week. Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program☺

#### **(90 pts) Implementation: Problem Statement**

**You will write a program to play blackjack using C++ classes.** Blackjack is a card game that has a dealer and 1 or more players who are trying to get a hand closest to 21 without going over. Aces can be 1 or 11, whichever is to your advantage, and all face value cards (Jack, Queen and King) have a value of 10.

The players begin with a specific amount of money, and only the players can bet against the dealer. First, everyone decides how much he/she wants to bet. Then, everyone is dealt 2 cards face up, and the dealer gets one face up and one face down. Each player decides whether to receive another card or not without going over 21. If the player goes over 21 (or busts), then he/she immediately loses their bet from their total playing money. After all players finish receiving cards, then the dealer turns over the card that is facing down, and the dealer must receive a new card if the total is below 17 and stay/hold if the total is 17 or above.

- The players with totals over 21 immediately lose their bet from their playing total.
- If the dealer goes over 21, then the players with 21 and under all win their bet, which is added to their playing total.
- If the dealer is 21 or under, then the players with card totals less than the dealer lose their bet from their playing total, and those players with card totals greater than the dealer win their bet, which is added to their playing total.
- Players who have the same card total as the dealer neither win nor lose, their playing total remains unchanged.
- Players who have a card total of 21, blackjack, win 1.5 times their bet to their playing total.

**You will be required to have the following classes and members. However, you may add more members and functions, based on your design. You will make .cpp files for each class, .h, interface file. Make sure you include the correct .h files with the .cpp files.**

#### **//card.h interface file**

```
class card {  
    private:  
        int value;    //1-13
```

```

        char *suit; //4 each: club, spade, heart, diamond
public:
    //must have constructors
    //must have destructors
    //must have accessor functions
    //must have mutator functions
};

```

#### **//deck.h interface file**

```

class deck {
private:
    card cards[52];
    int num_cards;
public:
    //must have constructors
    //must have destructors
    //must have accessor functions
    //must have mutator functions
};

```

#### **//hand.h interface file**

```

class hand {
private:
    card *cards;
    int num_cards;
public:
    //must have constructors
    //must have destructors
    //must have accessor functions
    //must have mutator functions
};

```

#### **//player.h interface file**

```

class player {
private:
    hand p_hand;
    int playing_total;
    int card_total;
    int bet;
public:
    //must have constructors
    //must have destructors
    //must have accessor functions
    //must have mutator functions
};

```

#### **//dealer.h interface file**

```

class dealer {
    private:
        hand d_hand;
        int card_total;
    public:
        //must have constructors
        //must have destructors
        //must have accessor functions
        //must have mutator functions
};

```

### **//game.h interface file**

```

class game {
    private:
        deck cards;
        player *players;
        dealer game_dealer;
        int num_players;
    public:
        //must have constructors
        //must have destructors
        //must have accessor functions
        //must have mutator functions
};

```

### **Your program must be able to:**

- Setup a deck of 52 cards with spades, clubs, diamonds, and hearts of 2-10, Jack, Queen, King, and Ace for each suit.
- Shuffle the deck of cards before playing any game.
- Determine how many players and the playing total for each.
- Make sure that a player's bet is less than or equal to the playing total.
- Determine when a full deck has been dealt, and reshuffle.
- You should not have any memory leaks in your program.

### **Extra Credit (10 pts):**

In blackjack, if a player starts with doubles (cards with the same value), then the cards can be split into different hands. The player is given two cards, one for each hand, and the dealer now gets two cards for a different hand. After the first hand is played, then the second hand is played between the dealer and the player who split.

(10 pts) **Program Style/Comments**

In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line! [http://classes.engr.oregonstate.edu/eecs/spring2016/cs162-001/162\\_style\\_guideline.pdf](http://classes.engr.oregonstate.edu/eecs/spring2016/cs162-001/162_style_guideline.pdf)

```
/******  
** Program: play_game.cpp  
** Author: Your Name  
** Date: 04/20/2016  
** Description:  
** Input:  
** Output:  
*****/
```

Electronically submit your C++ program (**.h, .cpp, and Makefile files**, not your executable!!!) and your test files as a tarred archive by the assignment due date, using TEACH.

**You must tar these files together using the following command:**

```
tar -cvf assign2.tar card.h card.cpp deck.h deck.cpp hand.h hand.cpp  
player.h player.cpp dealer.h dealer.cpp game.h game.cpp play_game.cpp  
Makefile
```

**\*\*NOTE:** The easiest way to upload your program from ENGR to TEACH is to map a network drive to your home directory on ENGR. Mac or Windows, See: <http://engineering.oregonstate.edu/computing/fileaccess/>