

Programming Assignment #7: Let's Play Blackjack!

Due Date: April 29, 1998

1 The Problem

In this program, you will use the card class along with new classes called deck and player to write a (simplified) Blackjack game.

The game is played between two players as follows. Each player takes three cards from the deck. They then add up the points in their respective hands. The point value of a card whose value is between two and ten is simply the value of the card. For example, the point value of the 'four of clubs' is four points. The value of a face card (i.e., Jack, Queen or King) is ten points, and the value of an Ace is eleven points. The player whose point total is closest to 21 without going over is the winner. If both players exceed 21 points or if they have the same point value, there is no winner. In addition, if exactly one of the players exceeds 21, the other player is the winner.

2 The Program

The files you will need for this program are contained in the folder H:\nomencl\assnj\prog7. There are three files in this folder: cards.h, cards.cpp and test.cpp. The file cards.h is the header file containing the class declarations for the classes card, deck and player. All the member function prototypes are contained in this file. The file cards.cpp contains the implementations of all the constructors and member functions. The instructor has implemented some of the constructors and member functions for the card and deck classes. You must implement all the constructors and member functions for the player class whose (partial) declaration appears

below.

```
class player {
private:
    card    hand[13];    //room to store 13 cards
    int     numOfCards; //number of cards currently
                        //in the hand
public:
    *****
};
```

A player is composed of a hand, an array `hand[13]`, of up to 13 cards and an integer `numOfCards` that specifies the number of cards currently in the hand. For the Blackjack game, you will only need three positions of `hand[13]`. The constructors and member functions you must code are as follows:

```
player()           //constructor
takeACard(deck& d) //member function
computePoints()   //member function
showHand()        //member function
```

See attached code listings for details. In addition, you must code a function called `whoWins(player p1, player p2)` that determines which player has won the game. You must also complete the coding of the main program. The main program should perform the following tasks:

- 2 Shuffle the deck 100 times before playing. To shuffle the deck, the program uses a random number generator. The instructor has coded this for you. All you have to do is call the member function `shuffle(int n)` with `n = 100`. When the program starts executing, it will ask the user for a positive integer to 'seed' the random number generator. Just type in any positive integer. Using the same seed will cause exactly the same hands to be drawn from one run of the program to the next. Therefore, change your seed values to generate different games!
- 2 Draw three cards from the top of the deck and place them in the hand of player 1.
- 2 Draw three cards from the top of the deck and place them in the hand of player 2.

- ² Print out the hand of player 1 and the point value of the hand.
- ² Print out the hand of player 2 and the point value of the hand.
- ² Print out the winner of the game.

3 What To Turn In

Turn in four files: cards.h, cards.cpp, test.cpp and test.exe. In addition to the disk, you should turn in a printout of each of these files (except test.exe). The disk and printouts should be placed in a two-pocket folder. On the front of the folder, the following information must appear.

Your Name Comp 170 Spring Semester 1998 Section 611 Instructor: Dr. John Del Greco

4 Extra Credit Embellishment

There is no extra credit embellishment for this program.

5 Late Program Policy

A program will lose 10% of its value each day it is late (excluding weekends). Starting early on your programs will maximize your chances of earning full credit on your work!