

# Programming Assignment 5

Computer Science 110-03

Due date: Wednesday, December 7

## 1 Blackjack

In the game of blackjack, one or more players play against the dealer. At the start of the game each player is dealt two cards face up. The dealer is dealt one card face up and one card face down. Then, starting with the first player, each player can get more cards (“hits”) until she “stands” or “busts.” The dealer is the last person to play. When she starts to play, she turns her face down card — her “hole” card — face up.

The object of the game is to get a total that’s as close to 21 as possible, without getting a total over 21. Each card with a numerical rank (2, 3, . . . , 10) is worth its numerical value, face cards are worth 10, and an ace is worth 1 or 11. (Suits don’t matter.) A total over 21 is a “bust.” A total of 21 in the first two cards is called a “natural” or “blackjack.”

A natural beats a score of 21 in three or more cards. Otherwise a score closer to (but less than or equal to) 21 wins. In most cases equal scores are ties, and neither the dealer nor the player wins. This also applies if both the dealer and the player have blackjack. The exception to the rule is that if both the dealer and the player go bust, then the dealer wins.

The dealer has no choice in how she plays her hand: if her total is 17 or higher, she must stand, and if her total is 16 or less she must hit. If she has an ace and one of the possible totals is greater than 16 but less than or equal to 21, then she must stand.

## 2 Program 5

Your program should begin by getting the number of players (including the dealer) from the user. It should then deal each player one card at a time until all the players (including the dealer) have two cards. After the initial deal is completed the program should show each player’s hand, except that it should hide the dealer’s hole card.

It should then begin with the first player and give him the option of hitting (‘h’) or standing (‘s’). This should be repeated until the player stands or goes bust. After each hit, the program should show the hands of all the players (including the dealer), and it should print the possible total(s) for the first player. This procedure is repeated for each player (other than the dealer).

When it’s the dealer’s turn to play, all the hands should be shown and the dealer’s hole card turned face up. Before playing the dealer’s hand the program should first check whether all the other players have gone bust. If so, the dealer wins against each player, and the program can terminate. If at least one player hasn’t gone bust, the dealer should begin play according to the rules described above. When the dealer finishes play, the program should print each player’s score, and who won each hand played against the dealer.

### 3 Some Details

Before beginning play, your program should create a deck of 52 cards and shuffle it. If, at any time, it's necessary to deal a card and the deck is empty, your program should just print a message and quit.

You should use the Python code `play_cards0.py` on the class website for some of the code: it has functions for printing a card, creating a deck, shuffling a deck, and creating a list of possible values for a hand.

### 4 Due Date

In order to receive full credit, your program must be in the `p5` subdirectory of your Subversion repository by 2:00 pm on Wednesday, December 7, and you must turn in a print out of your program by 5 pm on the 7th.

### 5 Grading

Your program will be graded on the basis of its correctness and its “static features.”

1. Correctness will be 60% of your grade. Does it correctly show player's hands? Does it correctly carry out the player's instructions? Does it handle busts correctly? Does it correctly evaluate each player's total?
2. The following static features will be graded.
  - (a) Documentation will be 10% of your grade. Does your header documentation include the author's name, the purpose of the program, and a description of how to use the program? Are the identifiers meaningful? Are any obscure constructs clearly explained?
  - (b) Source format will be 5% of your grade. Is the indentation consistent? Have blank lines been used so that the program is easy to read?
  - (c) Quality of solution will be 20% of your grade. Does your solution contain unnecessary calculations? Is your solution too clever — e.g., has the solution been condensed to the point where it's incomprehensible? Are functions and data structures (e.g., lists, strings, dictionaries) used properly? Are any functions (including the main program) more than 20 lines long?

### 6 Collaboration

It is OK for you to discuss solutions to this program with your classmates. However, no collaboration should *ever* involve looking at one of your classmate's source programs! It is usually extremely easy to determine that someone has copied a program, even when the individual doing the copying has changed identifiers and comments. If we discover that someone has copied a program, the authors of *both* programs will receive an F in the course.