

**CSS143 – Spring 2005**  
**Assignment #4 (Blackjack)**  
**Assigned 05/24, due 06/02 at 4:15PM**

For this assignment you will write an interactive text-based blackjack game. I assume you have basic knowledge of the rules of blackjack. If not, you will find ample opportunities to learn (and lose some money) on the web. My personal favorite (for learning) is <http://www.blackjackinfo.com/>

Your game will proceed as follows:

- On initialization the player will be asked for their “bank amount,” the total amount of money they want to play with. The game ends when the player runs out of money, or explicitly asks to terminate the game at the end of a hand. The player is also asked for the number of decks to play with – acceptable values are 1 through 8. These cards are put in a "shoe" from which a single card can be dealt.
- After cards are put in the shoe, they are shuffled.
- Then the end of the shoe is marked: a random number is chosen that is at least  $\frac{3}{4}$  of the total number of cards in the shoe, but at most  $\frac{15}{16}$  of the total cards in the deck. The shoe is marked at that point, and play is interrupted when all cards to that point have been played (actually at the end of the current hand). At that point the shoe is renewed, re-shuffled, and re-marked.

Then we play “hands” until the player requests termination, or runs out of money.

A hand proceeds according to the rules of blackjack

1. the player bets an amount of money not to exceed the current bank balance
2. the player is dealt two cards and the dealer is dealt two cards, only one of which is visible to the player
3. if both player and dealer have blackjack, this is a “push” and nothing happens
4. if the dealer has blackjack, the hand ends and the bet amount is subtracted from the player’s bank
5. if the player has blackjack, the player is paid 150% of the bet amount
6. the player can request additional cards but if the hand total exceeds 21 points, the hand ends and the player loses the bet money
7. the dealer then takes cards until the point score is at least 17 (except the dealer must hit a “soft 17”)
  - a. if the point total exceeds 21 points, the player is paid 100% of the bet amount, otherwise
  - b. if the point total exceeds the player’s point total, the player loses the bet amount, otherwise
  - c. if the point totals are equal, nothing happens, otherwise
  - d. the player is paid 100% of the bet amount

Again, let the web be your guide to filling in the details (like how to score a hand).

At the end of every hand, the player is told the current bank balance, and given the opportunity to quit.

---

### **Requirements for the Program**

You will be given a working version of the program, and also "shells" for all the classes (basically the public interfaces and constructors, but some method code in some cases). You must use these shells in writing your solution. Also, for consistency please use the version of iutilities in the assignment folder.

Duplicating the functionality of the working version is the "base version" of the game. Consider extending the game for extra credit.

---

### **Extra Credit**

Here are some ideas – feel free to propose your own. These are listed in order from easiest to hardest (I think)

1. Surrender: after the initial deal, the player has the option of surrendering half the bet amount
2. Doubling down. Allow the player to double down
3. Split pairs. Allow the player to split pairs. By convention, when you split aces you only get one additional card, but when you split anything else you can request additional cards.
4. Insurance: Allow the dealer to offer insurance when showing an ace
5. Strategy advisor: there are some standard guidelines for playing the game (check the URL above). Have the program notify the player when a decision is in violation of those guidelines
6. Strategy advisor with card counting: there are some more complicated guidelines for playing the game assuming you are counting cards played. Implement an advisor that notifies the player when a decision violates these guidelines
7. Auto-play: write code that acts as the player. At the beginning you should ask how many hands to play, and continue play for that many hands, or until the auto-player goes broke.
8. Multiple players

---

### **What to Hand In**

1. Your source files, but not the ioutilities
2. The standard writeup, including details about whatever extra credit you did