CPSC 230: Intro to Computer Science

Blackjack Extra Credit Assignment

Armed with your knowledge of multiple Python concepts, you can now build a command-line-based interactive game: Blackjack (AKA 21). You will be using most of the concepts we have covered in class so far. Your program will probably include: *while* loops, *for* loops, conditionals, lists, (maybe) dictionaries, indexing, string operators, and functions. You are not required to build the program in any specific form, aside from making use of functions where they are useful (i.e. where you have repetitive code).

Rules of the game:

Blackjack is played with a standard deck of 52 cards: 2 – 10, Jack, Queen, King, and Ace of all four suits. The goal of the game is for the player to have a hand total as close to 21 as possible while not going over. The player and dealer start with two cards, and all cards except for one of the dealer’s cards are face-up. If the player’s hand makes 21 with the first two cards, then the player wins unless the dealer also has 21. If the dealer also has 21, the game is a tie (called a push). The player is the first to act, choosing to either **stand** (keep the same total), **hit** (take another card from the deck), **double** (take another card from the deck, but no moves after the double are allowed), and **split** (create two hands from one if you have 2 cards with the same number). If the player’s card total is above 21, the player automatically loses (this is called a bust).

Once the player busts, stands, or doubles, it is then the dealer’s turn. The dealer flips over their face-down card. If the dealer has 21, the player ties if they also have 21, and loses otherwise. In this version of the game, the dealer will continue taking cards (hitting) until they have a have a total of 17 or greater. For those of you that know the game well, this version has the **dealer stand on a soft 17**. Once the dealer has finished dealing cards to the player and to themself, they compare the hand totals. The person with the higher total wins the hand. Once the winner is determined, another round is played from the deck. For example, if the player has 18 and the dealer has 20, the dealer wins the hand and the process repeats again.

Notes:

Aces can act as 1 or 11, given the situation. If the player or dealer has an ace in their hand and the total is above 21 (counting the ace as 11), then the ace’s value becomes 1 so that the possessor of cards does not bust.

Implementing splitting of hands is optional. You will receive 2 extra points if you properly implement this feature.

Your program will have you play as one player against a dealer. You do not need to implement a betting feature.

Have fun!