# Extra-Credit Project 3: Blackjack (20 points)

## Objectives:

- Practice with dictionaries, conditionals, and functions
- Review concepts from entire class

## Play Blackjack Against the Computer

Write a program **`blackjack.py`** that allows a human player to play against a computer "house". You can find the rules for Blackjack here.

Create a normal deck of 52 playing-cards. Represent the deck as a dictionary in which each item is a card. The key for a card is a string representing the card type, e.g. , "AS" for Ace of Spades, "QH" for Queen of Hearts, and "7D" for Seven of Diamonds, and "9C" for 9 of Clubs. The value for a card is the card's points. Numbered cards are worth their numerical value, e.g., a "2D" is worth 2 points, so its card-item in the dictionary is "2D:2". Face cards (e.g., Jack, Queen, King) are worth 10 points. For simplicity, in this project an Ace is worth 11 points.

Start a game by initializing the player's hand and the computer's hand. Using the **`random`** module, deal two cards from the deck to the player and two to the computer. The player and the computer should each now have two cards. *Note*: Do not worry about duplicate cards, assume there are many decks in the shoe.

Call the function **`check_game_over`** (see below) to see if the player or the computer has won. If neither has won on the initial two-card deal, both the player score and the computer score are less than 21.

Now, ask the player if they want another card. If so, deal them a new card. Call **`check_game_over`** to see if the player's score has exceeded 21. Keep asking the player if they want another card until they say "no" or their hand exceeds 21 points. If the player's hand is over 21 points, the player has lost.

If the player is done getting cards and has not lost, it is the computer's turn. As long as the computer's score is less than 16, it **must** take another card. It **keeps taking cards** until its score is greater than 16. After each card, call **`check_game_over`** to make sure the computer's score is not greater than 21. If the computer's score exceeds 21, the computer loses and the player wins.

After both the player and the computer stop taking cards:

- If the player's score is greater than the computer's score, the player wins. Display the score and a message that the player won.
- If the computer's score is greater than the player's score, then the computer wins. Display the score and a message that the computer won.
- If the computer's score is equal to the player's score, then display that it is a tie.

Your program **must** include the following functions:

- **main():** the main function that is the backbone of the program.
- **deal_card(deck):** Given a deck of cards (see above), randomly choose and return a card (i.e., a key/value pair).
- **check_game_over(player_hand, computer_hand):** Given the player's and the computer's hands, check if either has won or lost the game, and return **`True`** if the game is over, and **`False`** if the game is not over. Here are the details:
  - If the player's score is 21 and the computer's score is less than 21, display a message "Blackjack! Player Wins!" and return **`True`** (meaning the game is over).
  - Similarly, if the computer's score is 21 and the player's score is less than 21, display a message "Blackjack! Computer Wins!" and return **`True`**.
  - If the player and computer both have 21, display a message "Double Blackjack! Tie." and return **`True`**.
  - If the player's score is greater than 21 **and** the computer's score is less than 21, the player loses and the computer wins! Display a message "Player broke 21. Computer wins!" and return **`True`**.
  - Similarly, if the computer' score is greater than 21 and the player's score is less than 21, the computer loses and the player wins! Display a message "Computer broke 21. Player wins!" and return **`True`**.
  - Otherwise, return **`False`** (meaning the game is **not** over).

## Extra Extra Credit (5 points)

Ask the user if s/he wants to play again and let them play another game against the computer without needing to restart the program.

## Submitting

Test and run the program on your laptop computer before submitting. Upload to Vocareum though Canvas Assignment.