



Blackjack

In this assignment, we will make a program to play the game of blackjack against the computer. In blackjack, the goal is to beat the bank by getting a total score as close as possible to (but not over) 21 points. An example of how this program would run is given below (user input in red):

Blackjack: beat the bank by getting as close to 21 as possible.
Ace can count as 1 or 11, and jack, queen and king are 10 points

You get 6 of clubs Your hand is worth 6

Draw another card (Enter) or stop giving('S + Enter'):<enter>

You get 10 of diamonds Your hand is worth 16

Draw another card (Enter) or stop giving('S + Enter'):<enter>

You get 4 of spades Your hand is worth 20

Draw another card (Enter) or stop giving('S + Enter'):S<enter>

Your total is 20

The bank plays:

Bank gets 3 of clubs Bank is worth 3

Bank gets 5 of spades Bank is worth 8

Bank gets 6 of hearts Bank is worth 14

Bank gets 8 of hearts Bank is worth 22

You won!

Phases of the game

The game consists of two phases: in the first phase the player picks random cards from the deck until (s)he is satisfied with the total score or if the total score exceeds 21. In the second phase, the bank (i.e., the computer player) picks random cards until either the score is larger than the score of the player, or until the score exceeds 21, whichever comes first. The figure below shows the steps in this program.

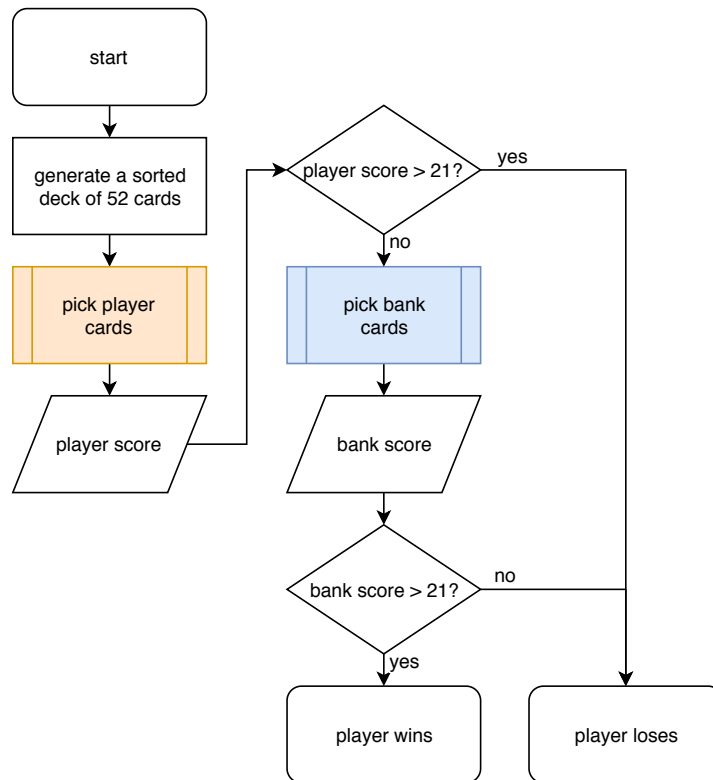


Figure 1: The main flow of the blackjack program

In the end, the program will need the following elements:

1. Code to generate a (shuffled) deck of 52 cards.
2. A function to determine the name of a card (e.g., 'three of diamonds').
3. A function to get the value of a card.
4. A function to calculate the total value of a hand of cards.
5. A function that lets the player compile a hand of cards.
6. A function to generate a hand of cards for the bank.
7. Code to determine who wins based on the total values of the players and the bank's hands.

1: Generate a shuffled deck of cards

To generate and store a shuffled deck of cards, we need to determine how we represent each card in our deck: do we store the name and value of each card? Or do we store just a sequence number, and determine the corresponding name and value when we draw a card?

In this assignment we will choose the second option: your code should generate a `list` containing the numbers 0-51. Use the `shuffle` function from the `random` library to shuffle the deck of cards:

```
import random
deck = ...
random.shuffle(deck)
```

2: Determine the name of a card

From the previous step, we have a list with the numbers 0-51, which represent the 52 cards in a standard deck of cards. Each of these cards is one of four suits (♦ diamonds, ♥ hearts, ♣ clubs, and ♠ spades). Each suit contains the cards ace, 2-10, jack, queen, and king. If we assume that the first card in a deck is the ace of diamonds, the second card the two of diamonds, and so on, and that every 14th card is a new suit, we can use *integer division*, and the *division remainder* to determine the name of a card for a given index. Check and complete the code below to see if you understand how the name of a card is determined from its index `card_idx`, when this index is a number between 0-51.

```
def name(card_idx):
    # Card indices are numbered 0 - 51. Every fourteenth card is a new suit
    suits = ["diamonds", "hearts", "clubs", "spades"]
    cards = ["ace", "2", "3", "4", "5", "6", "7", "8", "9", "10",
             "jack", "queen", "king"]

    isuit = ... your code here ...
    icard = ... your code here ...
    txt = cards[icard] + " of " + suits[isuit]
    return txt
```

Test your code with a couple of card indices. For instance, `print(name(0))` should print `'ace of diamonds'`, and `print(name(48))` should print `'ten of spades'`:

```
>>> print(name(0))
ace of diamonds
```

3: Get the value of a card

In blackjack, each card has a certain value. The numbered cards have a value equal to the number on the card, and the jack, queen, and the king are all worth 10 points. The ace is special: it can be either 1 or 11 points, whichever brings the total hand score closer to (but not over) 21 points. Assume for now that an ace is worth 1 point, and create a function `value(card_idx)` that returns card value based on card index:

```
def value(card_idx):
    val = ... # your code to determine the card value from its index
    return val # Value is a number between 1 and 10
```

4: Calculate the total value of a hand

As described above, calculating the total value of a hand is not as simple as just summing the values of all cards, since aces can be worth either one or 11, depending which of these options brings the total value closer to (but not greater than) 21 points. The value of an ace therefore depends on the rest of the cards in a hand.

Make a function `total(hand)` that has as input a hand of cards (i.e., a `list` of one or more card indices), which returns the total value of the hand of cards. Use the function from the previous step to determine the value of each individual card in your function:

```
def total(hand):
    # Iterate over the cards in hand, and calculate the total value.
    return tot # Additional question: What happens if you use the variable name
               ↳ "value" instead of "tot" here?
```

5: Let the player compile a hand of cards

Now that you have a deck of cards, and you are able to calculate the worth of a hand of cards, it's time to compile a hand of cards for the player and for the bank. This part of the code brings together all of the

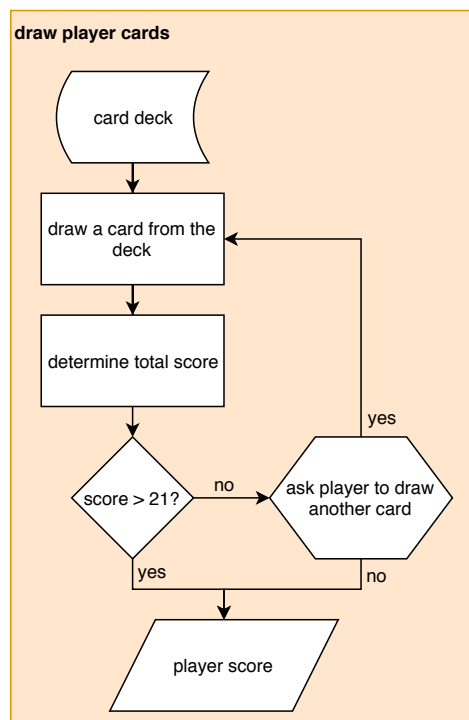


Figure 2: Steps to generate the player hand

functions you've made so far: you need to draw cards from the shuffled deck and show them to the player, together with the calculated total value of the current hand. Your code needs to repeat these steps until either the player wants to stop, or until the total hand value exceeds 21. This means that you need to use a loop (`while` or `for`), and that you need to get input from the player using `input()`. The resulting output when running this code should be as described in the beginning of this assignment. Make sure your code keeps the final score of the player.

6: compile a hand of cards for the bank

After the player is done, the bank gets to draw cards. The difference between a player and the bank is that the bank only stops drawing cards when either its score is higher than that of the player, or when its score exceeds 21. This makes the code to compile the bank hand easier, as you don't need the `input()` statement.

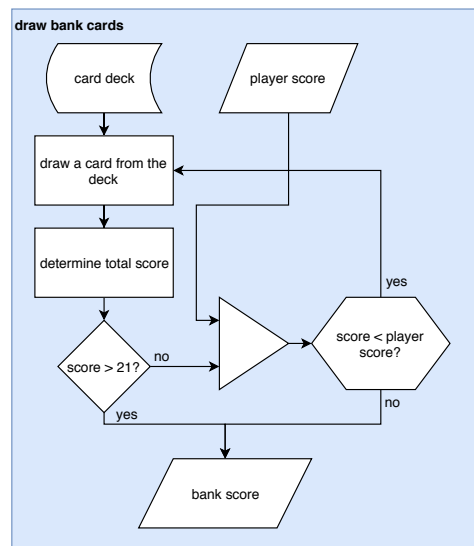


Figure 3: Steps to generate the bank hand

7: calculating the outcome of the game

Now you should have enough to generate a total score for both the player and the bank. There are two situations where the bank wins (player score > 21 or player score < bank score), and one situation where the player wins (bank score > 21). Check for these outcomes and print the result.

Extend the game for multiple players

Finding it hard to let this assignment go? Understandable! If you want you can extend the game for multiple players. You can start by asking how many players will join, and ask for their names. You will have to keep score for each player. The bank plays when all players are done. All players play only against the bank. Hands aren't compared between players.

Enter the casino: allow betting on games

Blackjack is a real casino game. And a game wouldn't be a casino game if there weren't any money involved! Now that you have multiple players, change your game so that each player starts with a sum of money, and can bet on each hand of cards. This means that you will have to play multiple games after each other, until you run out of money! The rules for betting are easy: at the start of each round, after you get your first hand of cards you can bet a sum of money. In the end, if you win from the bank you will get back twice your bet.

Completing the game

If you've been to a casino before you might have noticed that we've cut a few corners with the Blackjack rules in this assignment (e.g., that the bank actually starts with a single card, what happens when you start with a hand value of 21 (blackjack), splitting a pair of the same cards). Have a look at Wikipedia and complete the rules of blackjack in your game!