

MCS 260 Project Three : a simplified blackjack game due Monday 22 October at 1PM

The goal of this project is to apply top down functional design to simulate a simplified blackjack game. In what follows below, the user of the program is called the player, and our friend the computer is named the dealer. The rules of our simplified blackjack game are

1. A random number between 1 and 11 is given to the dealer and player in each round.
2. No more cards are dealt if the value of the hand of either dealer or player is 21 or more.
3. The player can choose to take another card or to the end the game.
4. The hand of the dealer is not shown until the dealing of cards has ended.
5. At the end of the game the hands of player and dealer are compared to see who wins.

A player whose value of the hand exceeds 21 is busted and we will call it a blackjack if the value equals 21. The rules to decide who wins are

1. At a tie, the hands of both dealer and player have the same value, or both are busted.
2. The player has a winning hand if its value is less than or equal to 21 and it is higher than the value of the hand of the dealer or the dealer is busted.
3. The player loses in all other cases.

The hand of the player must be shown, along with its sum, each time before before asking whether another card is wanted. At the decision of the outcome of a game, both hands of player and dealer are shown, along with their sums. Besides printing `win`, `tie`, or `loss`, also `busted` and `blackjack` must be announced whenever appropriate.

After reporting the outcome of each game (win, tie, or loss), the user is prompted the question `another game? (y/n)` and the program then continues if 'y' answers the question. At the very end of the program, the total tally of wins, ties, and losses is printed to the screen. A sample session of the program is at the back of this sheet.

Some important points:

1. One of the goals of this project is to use functions.
Correct programs without the use of functions can only get half of the points.
2. Your top down design should be apparent in the structure of your program.
Providing appropriate documentation will receive proper credit.
3. Handing in an incomplete but working program is better than handing in a program that crashes or does not run at all.
4. The first line of your Python program must be
`# MCS 260 Project Three by <Author>`
where you replace the `<Author>` by your name.
5. Email your solution to the project to `jan@math.uic.edu` before 1PM on Monday 22 October so the date of the email is proof of an on time submission. As a backup, bring also a printed version of your solution to class.

If you have questions or difficulties with the project, feel free to come to my office for help.

As a guide with formatting dialogues, below is a sample session with the program `sbj.py`, executed at the command prompt `$`:

```
$ python sbj.py
welcome to our simplified blackjack game
hand = [5] sum = 5
another card ? (y/n) y
hand = [5, 8] sum = 13
another card ? (y/n) y
dealers hand = [7, 10, 7] sum = 24 busted
players hand = [5, 8, 11] sum = 24 busted tie
another game? (y/n) y
hand = [5] sum = 5
another card ? (y/n) y
hand = [5, 11] sum = 16
another card ? (y/n) n
dealers hand = [2, 1] sum = 3
players hand = [5, 11] sum = 16 win
another game? (y/n) y
hand = [2] sum = 2
another card ? (y/n) y
hand = [2, 11] sum = 13
another card ? (y/n) y
hand = [2, 11, 4] sum = 17
another card ? (y/n) y
dealers hand = [1, 6, 4, 7] sum = 18
players hand = [2, 11, 4, 6] sum = 23 busted loss
another game? (y/n) y
hand = [6] sum = 6
another card ? (y/n) y
hand = [6, 2] sum = 8
another card ? (y/n) y
dealers hand = [5, 9, 9] sum = 23 busted
players hand = [6, 2, 2] sum = 10 win
another game? (y/n) y
hand = [3] sum = 3
another card ? (y/n) y
hand = [3, 6] sum = 9
another card ? (y/n) y
dealers hand = [11, 6, 4] sum = 21 blackjack
players hand = [3, 6, 10] sum = 19 loss
another game? (y/n) y
hand = [9] sum = 9
another card ? (y/n) y
hand = [9, 8] sum = 17
another card ? (y/n) y
dealers hand = [2, 3, 8] sum = 13
players hand = [9, 8, 4] sum = 21 blackjack win
another game ? (y/n) n
#wins = 3, #ties = 1, #losses = 2
$
```