Computer Science — Python — HW #9

Assigned on Wed, 2017-03-01. Due on Tue, 2017-03-07.

1. Read chapter 8 and 9 of Think Python, 2nd ed. (Continuing with two chapters per week.)
2. Lambdas, map, and filter. Recall that

**map**(f, items) behaves like [ f(x) for x in items ]

and that

**filter**(p, items) behaves like [ x for x in items if p(x) ]

Figure out the output of the following script. Then run the script to confirm your understanding.

from random import randint

values = [ 'A','1','2','3','4','5','6','7','8','9','10','J','Q','K' ]

suits = [ 'C', 'D', 'H', 'S' ]

deck = [ (value, suit) for value in values for suit in suits ]

positions = [ randint(0, 51) for k in range(0, 7) ]

cards = [ deck[pos] for pos in positions ]

to\_value = **lambda** c: c[0]

to\_suit = **lambda** c: c[1]

def print\_card(c):

print('{0:s} of {1:s}'.format(c[0], c[1]))

for c in cards:

print\_card(c)

print('=====')

for h\_card in **filter**(**lambda** c: to\_suit(c) == 'H', cards)

print\_card(h\_card)

print('=====')

for v in **map**(to\_values, cards)

print('Value: {0:s}'.format(v))

print('=====')

for s in **map**(to\_suits, cards)

print('Suit: {0:s}'.format(s))

1. **[Turn in]** Course project brainstorming. Submit descriptions of three candidate ideas for your course project. The descriptions should be "elevator pitches" — brief enough to be spoken in a brief elevator ride, but long enough to convey an idea of what the project would entail and possibly what makes it interesting. If it were to be an elevator pitch for a movie, this amount of detail would be OK:

The year is 1895.  In an inventors' laboratory co-op, just after the Frankenstein monster is brought to life, he accidentally walks into the active portal of a just-invented time machine next door.  In the distant future, he meets up with a super-intelligent talking dog named Quetzal. Opens in theaters July 5, 2019.

**Note:**

* 1. You are welcome to collaborate with others in the course, but there must be a clear separation/demarcation of responsibilities within each team: Who is going to design and write which portions of the final project.
  2. Your project doesn't need to be written from scratch. There are plenty of Python libraries and frameworks that are publicly available. For example:
     + Tkinter is a "windowing toolkit" that can be used to create a desktop app. We'll be briefly covering this in Unit #3 of the course. (See <https://docs.python.org/3/library/tk.html>)
     + Pygame is a set of Python modules for writing video games ([www.pygame.org](http://www.pygame.org)).
     + If you search online for "python chat client", you can find some sample programs that could form the basis of inter-computer communications.
  3. Don't worry too much about the scope of the project right now. Small ideas can be elaborated and enriched as needed, while overly ambitious ideas can be pared down.
  4. We'll have a couple more homework assignments in Unit #2 focused on brainstorming, so this isn't the final chance to come up with a project idea.
  5. In Unit #3 of the course, you'll create a project plan for your course project. That might involve some research, including writing one or more small proof-of-concept programs. Even then, you can change the project, though it will take more effort than settling on a project in Unit #2.