Quiz 2

To get credit for this quiz, use the Quiz tool at eee.uci.edu to enter your answers, within the Sunday-to-Tuesday quiz period.

Problem 1 (4 points)
For each of these sequences of statements, what does Python print?

(a) \( p = [2, 4, 6, 8] \)
\( \text{print}(p[0] + p[2]) \)

(b) \( \text{Restaurant} = \text{namedtuple}(\text{'Restaurant'}, [\text{'name', \text{'cuisine', \text{'phone', \text{'dish', \text{'price'}}})} \)
\( \text{fancy} = \text{Restaurant}(\text{'Taillevent'}, \text{'French'}, \text{'01-11-22-33-44'}, \text{'Escargots'}, 55.00) \)
\( \text{fast} = \text{Restaurant}(\text{'McDonald's'}, \text{'Burgers'}, \text{'334-4433'}, \text{'Big Mac'}, 3.95) \)
\( \text{print(fast.name, \text{' serves'}, fast.cuisine) \)
\( \text{print(\text{'True or False:'}, fancy.price > fast.price) \)

Problem 2 (6 points)
Assume you have the following definitions:
\( a = 5 \)
\( \text{Animal} = \text{namedtuple}(\text{'Animal'}, [\text{'name', \text{'species', \text{'age', \text{'weight'}}})} \)
\( L = ['Tinker', 'Evers', 'Chance'] \)

What is the data type of each of the following expressions? Choose from int, float, bool, str, list of __________ (specify), or Animal.

(a) \( a * 10 \)

(b) \( 3.14159 \)

(c) \( 'rhinoceros' \)

(d) \( 12 * (5 + 1) \)

(e) \( \text{len('Hippopotamus')} \)

(f) \( \text{Animal('Roger', 'rhinoceros', 45, 1500)} \)

(g) \( \text{Animal('Roger', 'rhinoceros', 45, 1500).age} \)

(h) \( [2, 4, 6, 8, 10] \)
(i) L

(j) L[1]

(k) [Animal('Roger', 'rhinoceros', 45, 1500),
    Animal('Harry', 'hippopotamus', 25, 1600),
    Animal('Eloise', 'elephant', 49, 4500)]

(l) [Animal('Marvin', 'marmoset', 5, 6).weight,
    Animal('Sally', 'siamang', 25, 15).weight,
    Animal('Lenny', 'lemur', 8, 4).weight,
    Animal('Casper', 'capuchin', 7, 10).weight]

Problem 3 (12 points)
The Anteater Grocery Store represents each item in its inventory with:

- a string representing the item's name, e.g., 'Granny Smith Apples 1 lb.'
- a float representing the item's price, e.g., 2.50
- an int representing how many of this item are in stock, e.g., 85

(a) (2 points) Define a namedtuple called Item to represent grocery items as described above.

Item = namedtuple('Item', 'name price in_stock')

(b) (2 points) Write a statement that assigns to the variable item1 an Item representing Campbell's Chicken Soup, selling for $1.25 per can, with 250 cans in stock.

item1 = Item('Campbell's Chicken Soup', 1.25, 250)

(c) (2 points) Write a Python expression for the value of the store's inventory of item1 (that is, how much money we'd take in if we sold all of that item we have in stock).

item1.price * item1.in_stock

(d) (4 points) Suppose we have this list of items:

L = [Item('pears', 2.50, 20),
    Item('plums', 3.25, 40),
    Item('oranges', 3.00, 35),
    Item('peaches', 2.50, 40)]

Write a Python expression representing the total value of the inventory of the first and last items on the list. For full credit, your expression should work for a list of any length greater than 1.

L[0].price * L[0].in_stock + L[-1].price * L[-1].in_stock
(e) (2 points) Fill in the blank with a Python statement that alphabetizes the list \( L \) of items (so that for the value of \( L \) in part (d), the print statement below would print \( \text{The first item is oranges} \)). (Your code should work to print the alphabetically first item in \( L \), no matter what items \( L \) contains.)

\[
\text{______________________________}
\]
\[
\text{L.sort()}
\]
\[
\text{print('The first item is', L[0].name)}
\]

**Problem 4** (5 points)

For each of these sequences of statements, what does Python print?

(a)  
```
def triple(n: int) -> int:
    ''' Return three times the parameter
    '''
    return n * 3

print('Four')
print(triple(5))
print('Score')
print(triple(2) + triple(10))
print(triple(3), triple(100))
print(triple(triple(4)))
```

(b)  
```
def print_n_copies(n: int, s: str):
    ''' Print specified number of copies of string
    '''
    print(n * s)
    return

print('Heads')
print_n_copies(4, 'Flip')
print('Tails')
```
Problem 5 (4 points)

In this Python code:

```python
def double(n: int) -> int:
    ''' Return twice the parameter value '''
    return 2 * n
print(double(13), "should be 26")
```

identify each of the following:

1. function name (in definition) — `double`
2. function definition — `def` line and next two lines
3. function call (of the function `double`) — `double(13)`
4. argument to a call to `double` — `13`
5. definition of a parameter in `double` — `(n: int)`
6. use of a parameter in `double` — `2 * n`
7. return type specification — `-> int`
8. docstring comment ("purpose statement") — triple-quoted string

(On a paper exam, you could circle portions of the code and draw arrows. Electronically, just copy the list and, after each item, copy the text that applies.)

Problem 6 (8 points)

(a) What are the advantages of dividing our code into functions?

- The same code can be called (reused) from many parts of the program
- We reduce clutter by avoiding duplicate code
- It provides interchangeable components (use the lemon icing/use the chocolate icing)
- Generally, it avoids clutter and keeps complexity under control by giving code segments a name; it's a form of abstraction (you can just call the function by name without having to know how it's implemented under the hood.

(b) Why is it a good idea to avoid duplicate code in our programs?

- Duplicate code makes the program longer than it needs to be. Unnecessarily longer code is more cluttered and harder for people to read and understand.
- If you need to change code that appears more than once, you have to find all the appearances and change all of them, or your program becomes an inconsistent mess.

(Lab Assignment 3 addresses this issue. It's important to read the explanatory material in the lab assignments.)

Problem 7 (1 point)

Yes or no: You got a score of “check” (3) on a lab assignment. Your roommate, who has a different TA, got a score of “check-plus” (4) and as far as you can tell (by comparing notes after submitting your work) your submission was about the same as your roommate’s. Should you be worried about your grade in the course because your TA seems to be a harder grader than your roommate’s TA?

No. As stated on the syllabus, final grades are determined separately, TA by TA, so students of a TA who gives more high scores won’t get a disproportionate share of the high grades. Perhaps you should be moderately concerned about why you got a check from your TA, though, since evidently your TA found something lacking in what you submitted. Check the comments section next to your score in the EEE GradeBook and if that doesn’t resolve it, check with your TA in person.