ICS 31 • UC Irvine
Fall 2016 • David G. Kay

Your Name ______________________________
Your Student ID (8 digits) ________________
Your UCInet ID ___________________________

Your Lab Section (circle one):

1.	8:00a	Syed Safir
2.	9:30	Syed Safir
3.	11:00	Nathaniel Baer
4.	12:30	Nathaniel Baer
5.	2:00	Yadhu Prakash
6.	3:30	Yadhu Prakash
7.	5:00	Anurag Mishra
8.	6:30	Anurag Mishra
9.	8:00p	Jason Desrosiers
10.	8:00a	Swarun Krishnamoorthy
11.	9:30	Swarun Krishnamoorthy
12.	11:00	Harun Anver
13.	12:30	Harun Anver
14.	8:00a	Karthik Prasad
15.	9:30	Karthik Prasad
16.	11:00	Jason Desrosiers

First Midterm

You have 75 minutes (until the end of the class period) to complete this exam. There are 55 points possible, so allow approximately one minute per point and you'll have plenty of time left over.

Please read all the problems carefully. If you have a question on what a problem means or what it calls for, ask us. Unless a problem specifically asks about errors, you should assume that each problem is correct and solvable; ask us if you believe otherwise.

In answering these questions, you may use any Python 3 features we have covered in class, in the text, in the lab assignments, or earlier on the exam, unless a problem says otherwise. Use more advanced features at your own risk; you must use them correctly. If a question asks for a single item (e.g., one word, identifier, or constant), supplying more than one will probably not receive credit.

Remember, stay cool! If you run into trouble on a problem, go on to the next one. Later on, you can go back if you have time. Don't let yourself get stuck on any one problem.

You may not share with or receive from anyone besides the instructor or TAs any information or materials during the exam. You may not use any electronic devices.

Please write your answers clearly and neatly—we can't give you credit if we can't decipher what you've written.

We'll give partial credit for partially correct answers, so writing something is better than writing nothing. But be sure to answer just what the question asks.

Good luck!

Problem 1 (12 points)
Problem 2 (10 points)
Problem 3 (15 points)
Problem 4 (4 points)
Problem 5 (4 points)
Problem 6 (3 points)
Problem 7 (7 points)
Total (55 points)

Problem 1 (12 points)

Use the following definitions in this problem:

```
event = "Election"
days_left = 21
parties = ['Democratic', 'Green', 'Libertarian', 'Republican']
```

(a) (5 points) What does Python print as it executes the following sequence of statements? (Write your answers in the blank space to the right of the code.)

```
print("Vote Nov. 8!")
print("Don't Forget!")

print(5 * 4 == 20)
print((days_left * 10) - 1)
print(days_left / 3)
print(days_left * 10 > 300)

print(event + '...' + "Day")
print(len(parties) + len(event))
```

(b) (4 points) What does Python print as it executes the following sequence of statements? (Write your answers in the blank space to the right of the code.) Remember zero-based indexing.

```
print(event[1])
print(event[-1])

print(parties[0])
print(len(parties[1]))

print("There are", days_left//7, "weeks until the", event)
```

(c) (3 points) Each of the following is a boolean expression (its value is either True or False). For each expression, give its value.

```
len(parties) != 6
'Peace and Freedom' in parties

'c' in 'Voting'
'c' in event
parties[0] > "American Independent"

len(3 * 'ok') == 3 * len('ok')
```

Problem 2 (10 points)

(a) (3 points) The Anteater Fruit Stand represents each kind of fruit it sells in a namedtuple called Fruit that has four fields: the kind of fruit, the name of the grower, the quantity they have in stock, and the price (per pound).

Which of the following defines a namedtuple that satisfies this specification? Circle each correct statement; the correct answer may include *one or more* of A, B, C, D, or E.

A. `Apple =  Fruit('apple ZotFarms 225 1.39')`

B. `Fruit = namedtuple('Fruit', 'name farm instock price')`

C. `Fruit = namedtuple('Fruit', 'kind of fruit, grower name, amt available, price per lb')`

D. `Apple = namedtuple('Fruit', 'apple', 'Zot Farms', 225, 1.39)`

E. `Fruit = namedtuple('Fruit', 'variety grower quantity price')`

(b) (3 points) Which of the following creates a Fruit object as a namedtuple following the description above, to represent 125 pounds of apples grown by Zot Farms, on sale for $2.50 per pound? Circle each correct statement; the correct answer may include *one or more* of A, B, C, D, or E.

A. `fruit1 = Fruit('apple Zotfarms 125 2.50')`

B. `fruit1 = Fruit.apple('Zot Farms', 125, 2.50)`

C. `fruit1 = orange.quantity * apple.price`

D. `fruit1 = Fruit('apple', 'Zot Farms', 125, 2.50)`

E. `fruit1 = Apple('Zot Farms', 125, 2.50)`

(c) (3 points) Which of the following statements reflects a price decrease of $0.50 for the Fruit object stored in `fruit1`? [You may assume any of the correct Fruit definitions above.] Circle each correct statement; the correct answer may include *one or more* of A, B, C, D, or E.

A. `fruit1 = Fruit(fruit1.name, fruit1.farm, fruit1.instock, fruit1.price - 0.50)`

B. `fruit1 = Fruit('apple Zot Farms 125 -0.50')`

C. `fruit1 = fruit1._replace(price = 2.00)`

D. `fruit1.price = fruit1.price - 0.50`

E. `fruit1 = fruit1._replace(price = fruit1.price - 0.50)`

(d) (1 point) Of the possible solutions in part (c) above, one is technically correct but of lower quality than the others. Which solution is lower quality and why? Choose the *one* response from A, B, C, or D that best describes the lower quality solution; don't circle more than one.

A. `fruit1 = fruit1._replace(price = fruit1.price - 0.50)` because underscores are hard to type

B. `fruit1 = Fruit(fruit1.name, fruit1.farm, fruit1.instock, fruit1.price - 0.50)` because it requires more typing than the others

C. `fruit1 = fruit1._replace(price = 2.00)` because it's inflexible; it's only right if the original price is $2.50

D. `fruit1.price = fruit1.price - 0.50` because it's the shortest

Problem 3 (15 points)

Professor Amanda Anteater represents the students in her class with a namedtuple defined as follows:

```
Student = namedtuple('Student', 'name ID midterm project final')
```

where the name is a string, the ID is an int, and the remaining fields are floats storing the scores (in the range 0 to 100) on the indicated items in the class.

(a) (2 points) In the function definition below, fill in each blank with one Python variable name, function name, method name, constant, or operator to satisfy the problem specification.

```
def total_score(s: Student) -> float:
    ''' Return the student's score in the class, with the midterm worth 25%,
        the project 35%, and the final 40%
    '''
    return s.midterm * _________ + s._________ * 0.35 _________ s.final _________ 0.40
```

(b) (4 points) What do the following statements print? [The arithmetic is easy to do in your head.]

```
s1 = Student('Zot, Zoe', 11223344, 100, 100, 100)
s2 = Student('Irvine, Irving', 55667788, 50, 50, 50)
print(s1.name, total_score(s1))
print(s2.name, total_score(s2))
```

(c) (2 points) In the function definition below, fill in each blank with one Python variable name, function name, method name, constant, or operator to satisfy the problem specification.

```
def project_score(s: Student) -> float:
    ''' Return the student's score on the project
    '''
    return ______________ . _______________
```

(d) (2 points) In addition to the function `project_score` defined above, assume you have also defined similar functions called `midterm_score` and `final_score` that return the values of those fields.

If Prof. Anteater has a list of 300 students called `SL`, fill in the blanks below to reorder `SL` by each student's final exam score, lowest to highest.

```
_______________.sort(key = _______________)
```

(e) (5 points) In the function definition below, fill in each blank with a single Python constant, operator, or identifier name (variable, function, attribute, method) to satisfy the problem specification.

```
def final_higher (SL: 'list of Student') -> 'list of Student':
    """ Return a list of those Students in the parameter SL whose final exam score is
         higher than their midterm score
    """
    result = [ ]
    for s in ____________:
        if s.____________  ____________ s.midterm:
            ____________.append(____________)
    return result
```

Problem 4 (4 points)

Suppose we have a list of Students called SL, as in the previous problem.

Match the four for-loops below (A through D) with the most accurate description below.

A.
```
for s in SL:
    print(s.name)
```

B.
```
for s in SL:
    print(s.name, total_score(s))
```

C.
```
for s in SL:
    print(s, total_score(s))
```

D.
```
for s in SL:
    print(s.name, s.total_score)
```

___ Print the name of each student with his or her overall score in the class .

___ Produces an error message about the improper use of total_score

___ Produces the largest volume of output

___ Print the names of the students, one per line

Problem 5 (4 points)

Identify the data type of each of the following expressions, using definitions that appear in this exam where appropriate. Choose from:

```
int    float    bool    str    list    Fruit    Student
```

A. `SL`

B. `SL[0]`

C. `fruit1.price`

D. `fruit1`

E. `fruit1.price > 10.00`

F. `SL[1].name`

G. `[3, 4, 5, 6, 7]`

H. `len(SL[5].name)`

Problem 6 (3 points)

What does the following code print?

```
def alpha (n: int, s: str) -> str:
    return beta(s) * n

def beta (x: str) -> str:
    return x + "*"

print("Election Day")
print(alpha(3, 'OK'))
print("Vote!")
```

Problem 7 (7 points)

(a) (3 points) What does the following code print out?

```
times = 4
for i in range(times):
    print(i, times)
print('Halloween')
```

(b) (4 points) What does the following code print out?

```
print('Here we go:')
for p in ['Carter', 'Reagan', 'Bush', 'Clinton', 'Bush', 'Obama']:
    if len(p) < 6:
        print(p)
print('The end.')
```

Problem 8 (0 points)

When you're done with the exam, follow these steps (so you don't disturb your classmates and so your exam gets turned in properly):

- Gather up all your stuff.
- Take your stuff and your exam down to the front of the room.
- Turn in your exam; show your ID if asked.
- Exit by the doors at the front of the room. Don't go back to your seat or disturb students who are still working.