ICS 31 • UC Irvine
Fall 2012 • David G. Kay

Your Name _____KEY_____
Your Student ID (8 digits) _____
Your UCInet ID _____

Your lab:
Section (1-10) _____
Time MWF at: 8 10 12 2 4 6
TA's name _____

First Midterm

You have 75 minutes (until the end of the class period) to complete this exam. There are 40 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.

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 any information or materials with classmates during the exam and 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	(5 points)
Problem 2	(5 points)
Problem 3	(8 points)
Problem 4	(6 points)
Problem 5	(3 points)
Problem 6	(4 points)
Problem 7	(6 points)
Problem 8	(3 points)
Total	(40 points)

Bd3hyR

Problem 1 (5 points) **Simple expressions**

What does Python print as it executes the following sequence of statements? Remember zero-based indexing.

```
print((10 + 3) * 2)          26 (1/2 point)
print(11 / 2)                5.5 (1/2 point)
x = 25
print(x * 2 == 50)           True (1/2 point)
print(x / 2 >= 50)           False (1/2 point)
a = 'Honda'
b = 'Audi'
print(a + b)                 HondaAudi (1 point; accept upper or lower case)
c = len(a) + len(b)
print(c)                     9 (1 point)
print(a)                     Honda (1/2 point)
print(a[1])                  o (1/2 point)
```

Problem 2 (5 points) **List operations**

Using the list `L`, what is the value of each of the expressions (a) through (e) below?

```
L = ['Ford', 'Chevrolet', 'Toyota', 'Nissan', 'Tesla']
```

(a) `len(L)` **5**

(b) `len(L[1])` **9**

(c) `L[2]` **Toyota**

(d) `L[4] + '***'` **Tesla***

(e) `L[1] == L[2]` **False**

SCORING: 1 point each

Problem 3 (8 points) **Namedtuples**

Anteater Autos represents each car in its inventory with:

```
Auto = namedtuple('Auto', 'manufacturer model year price')
```

(a) (4 points) Write a statement to create a new Auto object that represents a 2005 Honda Accord whose price is $10,000, and to assign that object to the variable `x`. (Honda is the car's manufacturer.)

x = Auto('Honda', 'Accord', 2005, 10000) # price may be float, 10000.00

(b) (4 points) Write a statement that assigns to `y` an Auto just like `x`, except that its price is $50 higher.

y = x._replace(price=x.price+50) # EITHER IS OKAY
y = Auto(x.manufacturer, x.model, x.year, x.price+50)

Problem 4 (6 points) **Lists of namedtuples**

Suppose that Anteater Autos has a Python list of 100 Auto objects, called `AL`, and this definition:

```
def Auto_price (a: Auto) -> float:
    ''' Return the price of the auto
    '''
    return a.price
```

Fill in each blank below with one Python variable name, function name, method name, constant, or operator, to satisfy the problem specification.

(a) (1 point) Sort the list by price, lowest to highest.

```
AL.sort(key=_____________)
```

Auto_price

(b) (3 points) Compute the average price of the first and last objects on the list.

```
average = (_____________[0].price + _____________[-1].price) _____________   _____________
```

average = (AL[0].price + AL[-1].price)/2 **SCORING: 1/2 point for each AL; 1 point for /, 1 point for 2**

(c) (2 points) Print the model of the last item on the list.

```
_____________ (AL[-1]._____________)
```

print(AL[-1].model)

Problem 5 (3 points) **Types**

In each of the following Python expressions or statements, indicate what *data type* belongs in the indicated place by choosing one of these data types: `int` `float` `bool` `str` `list` `Auto`

SCORING: 1/2 point each

In some cases, more than one answer may be possible; just give one of the correct answers.

(a) `s = _________ + 17` **int or float (or bool, really)**

(b) `t = _________ + 'pie'` **str**

(c) `x[ _________ ] = 'catfood'` **int**

(d) `_________.sort()` **list**

(e) `if _________ :` **bool**

(f) `print(_________.price)` **Auto**

Problem 6 (4 points) Functions and parameters

In the function definition below, fill in each blank with one Python variable name, function name, method name, constant, or operator.

```
def adjusted_price(a: Auto, cutoff: float, amount: float) -> float:
   ''' If Auto's price is greater than the cutoff, increase it
       by the specified amount; otherwise increase it by $25.00.  Return
       the the changed price.
   '''
        if _______________.price > _______________:
                  a                      cutoff
                              return _______________.price + _______________
                                            a                    amount
        else:
                              return _______________.price + _______________
                                            a                    25.00
```

SCORING: 1 point for all three a's; 1 point for cutoff; 1 point for amount; 1 point for 25.00

Problem 7 (6 points) for loops

Using the list defined below, what does Python print when these statements are executed?

```
L = ['Ford', 'Chevrolet', 'Toyota', 'Nissan', 'Tesla']
```

(a) (2 points)

```
for x in L:
    print('Item:', x)
```

Item: Ford
Item: Chevrolet
Item: Toyota
Item: Nissan
Item: Tesla

(b) (4 points)

```
size = len(L)
for i in range(size):
    print(i+1, L[i])
print("Total count:", size)
```

1 Ford
2 Chevrolet
3 Toyota
4 Nissan
5 Tesla
Total count: 5

SCORING: Be forgiving about extraneous quotation marks

Problem 8 (3 points) **Parameter passing**

What does the following code print?

```
def first(s: str, i: int) -> str:
    return s * second(i)

def second(n: int) -> int:
    return n+3

print(first('*', 2))
```

*****