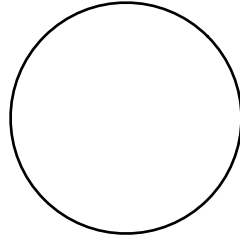


CSCI 1300 Exam 2 Spring 2013

Python Lists, Loops,
Dictionaries, Classes,
Objects



Write Clearly!

Name: _____

9-Digit Student ID: _____

TA (circle one):

Frank
Jing

Halley
Mahnaz

Jaeheon
Don't Know

1. Type Question (5 points — 1/2 point per item)

**Literal or
Expression**

Circle the appropriate type for the entire left cell

611	<u>integer</u> — float — string — boolean — None — dictionary — list — class — object
"611"	integer — float — <u>string</u> — boolean — None — dictionary — list — class — object
{ 'a': 611, 'b': "611" }	integer — float — string — boolean — None — <u>dictionary</u> — list — class — object
class Foo: pass	integer — float — string — boolean — None — dictionary — list — <u>class</u> — object
foo = Foo()	integer — float — string — boolean — None — dictionary — list — class — <u>object</u>
[foo, False, 'Hello!', 4+2]	integer — float — string — boolean — None — dictionary — <u>list</u> — class — object
611 > 30	integer — float — string — <u>boolean</u> — None — dictionary — list — class — object
5.0 / 2.0	integer — <u>float</u> — string — boolean — None — dictionary — list — class — object
"5.0" + "2.0"	integer — float — <u>string</u> — boolean — None — dictionary — list — class — object
5 / 2	<u>integer</u> — float — string — boolean — None — dictionary — list — class — object

2. Question on Types, Loops, and Lists (4 points)

```

1 things = ["Seven", 4.0, 3, True, 4.5, 10]
2 result = 0
3 for x in things:
4     if type(x) is int:
5         result = result + x
6 print result

```

A literal is exactly one type. E.g., if it is a float, it is not an int. What does line 6 print out?

13.
(only integers in the list are 3 and 10)

3. Loop Question (4 points)

```
8     x = 5
9     num = 0
10    while x < 50:
11        x = x * 3
12        num = num + 1
13    print "x:", x, "num:", num
```

What does line 13 print? Use the blanks below to tell us what it prints out for *x* *and* for *num*.

x: **135** num: **3**

4. Dictionary Question (4 points)

```
15    words = { "Guitar" : "A musical instrument",
16              "Ice"      : "Water that is too cold",
17              "Penguin" : "The Linux mascot" }
18    print "A Guitar is", words['guitar']
19    _____ = "Frozen water"
```

(a) There's a bug on line 18. When we run the program, Python complains with *"KeyError: 'guitar'"*. First, what does this mean (1 point)? Second, what should we do to line 18 to fix the problem (1 point)?

KeyError means the key 'guitar' is not found.
We should instead use 'Guitar' with upper case 'G'.

(b) What should go in the blank on line 19 so we re-define 'Ice'? (2 points)

words['Ice'] = "Frozen water"

5. Functions (5 points)

Complete the function below. It should print out a numbered list of key/value pairs, followed by the total number of keys. Sample output is shown to the right. There are *five blanks*, one point each.

```
1: Guitar = A musical instrument
2: Ice = Frozen water
3: Penguin = The Linux mascot
-----
3 key(s) total
```

```
# parameter 'stuff' is a dictionary
def summarize_dictionary(stuff):
    idx = 1
    for key in stuff:
        print str(idx) + ": " + key + " = " + stuff[key]
        idx = idx + 1
    print 20 * "-"
    print len(stuff), "key(s) total"
```

6. Class/Object Question (8 points).

```
class Widget:
    spokes = 0
    teeth = 0

    def __init__(self, spokes, teeth):
        self.spokes = spokes
        self.teeth = teeth
```

Widget ID	#spokes	#teeth
1	10	15
2	19	19
3	8	5

(a) Write code that does these two things:

1. Creates an empty list called `widgets` (1 point)
2. Adds three `Widget` objects into that list. (3 points) Widget specs are given in the table above.

```
widgets = [ ]
widgets.append(Widget(10, 15))
widgets.append(Widget(19, 19))
widgets.append(Widget(8, 5))
```

(b) (4 points) Central command has given us new requirements. `Widget` instances must have:

1. At least as many spokes as teeth (1 pt)
2. At least eleven teeth (1 pt)
3. An odd number of teeth (1 pt)

Complete the following method for the `Widget` class. It should check if the above rules are followed. If any of the requirements are not being followed, it should return `False`, otherwise it returns `True` (1 pt). *Be sure to use 'self' appropriately and to write clearly. You'll lose points if you don't. We will be sticklers on this.*

```
def check_correctness(self):
if self.spokes < self.teeth:
    return False
if self.teeth < 11:
    return False
if self.teeth % 2 == 0:
    return False
return True
```

```
# Another way of doing it:

if self.spokes < self.teeth or \
   self.teeth <= 11 or \
   self.teeth % 2 == 0:
    return False;
else:
    return True
```