This was the final CS1 exam given in May, 2014. I eliminated one question concerning classes and objects, and clarified a few points that were unclear in the original version. I also translated some things from Python 2 to Python 3.

1. Find the type and value of each of the following expressions. Warning: several of these expressions will result in an error.

(a) 5/ / 2 * 2.0
(b) 'three'+'3'
(c) 'three'* 3
(d) sorted([4,1,2,3])
(e) [4,1,2,3]+[4,1,2,3].sort()
(f) 3 in [4,1,2,3]
(g) 4 % 2 != 5/ /6
(h) [x**2 for x in range(10) if x%3==0]

2. Consider the function f defined below:

```python
def f(m):
    if m<1:
        return False
    else:
        while m%3 == 0:
            m=m/3
        return m==1
```

(a) Find f(0), f(1), f(2), f(3), f(9) and f(12).

(b) Describe in general what f (m) is telling you. (A good way to answer this question is to devise a descriptive name for the function f.)
3. The following three functions perform the same operation on lists, but they go about this task in different manners.

```python
def ext1(x,y):
    for j in y:
        x.append(j)

def ext2(x,y):
    for j in y:
        x.append(j)
    return x

def ext3(x,y):
    for j in y:
        x=x+[j]
    return x
```

Tell what output is printed by the following sequence of statements, assuming `ext` is replaced by `ext1`, `ext2`, and `ext3`. That is, you will give three different answers, each with two lines of output. By the way, this code does not produce any error messages, and every print statement produces some output.

```python
u=[2,3,5,7]
print (ext(u,[4,1,5]))
print (u)
```

4. Consider the function `g` defined below.

```python
def g(s):
    v=['']
    for c in s:
        v=v+[u+c for u in v]
    return v
```

(a) What is `g('do')`?
(b) What is `len(g('dog'))`? You don’t need to write out `g('dog')`, just figure out its length.
5. Write a function h that takes as a parameter a list and returns a list of all the 'suffixes' of the list in order of increasing length. By a suffix, I mean a sublist starting at some element and continuing to the end. So, for example,

\[ h([3,4,2,-9,7,6,1]) \]

returns

\[ [[]],[1],[6,1],[7,6,1],[-9,7,6,1],[2,-9,7,6,1],[4,2,-9,7,6,1],[3,4,2,-9,7,6,1]] \]

There are many different ways to do this, including one that has only a single line, and another that uses recursion.

6. Write a function sim_coin(m) that simulates the toss of m fair coins and returns the number of heads. You may assume that the statement import random has already been executed. You can use either random.randint(a,b), which returns a random integer between a and b inclusive, or random.random(), which returns a random float between 0 and 1.

7. Write a fragment of code that uses result of problem 6 to simulate 1000 tosses of 10 coins. The result of executing the code should be the creation of a dictionary d that records the number of times 0 heads came up, 1 head came up, etc. For example, after one run of this fragment, I printed the value of d and got the following output.

\{0: 1, 1: 12, 2: 44, 3: 118, 4: 194, 5: 240, 6: 213, 7: 128, 8: 36, 9: 12, 10: 2\}

This tells us that in 213 of the one thousand tosses, 6 coins came up heads. Of course, the result will be different each time you run the program. Note that you ought to be able to do this problem even if you did not complete problem 6; just call a function sim_coin that behaves as described in 6.
8. Consider the function \( \text{rptsquare} \) defined below.

```python
def rptsquare(e, b):
    if e==0:
        return 1
    else:
        u=rptsquare(e//2, b)
        if e%2==0:
            return u*u
        else:
            return u*u*b
```

(a) What value does it return when \( b=2 \) and \( e=0, 1, 2, 3, 4 \)? (The best way to do this is to consider each of these values of \( e \) in order, using the result of each part to find the answer to the subsequent part.)

(b) Give a succinct description of what \( \text{rptsquare} \) returns in general when \( e \) and \( b \) are integers and \( e \) is nonnegative.

(c) What happens when \( b=2 \) and \( e=-1 \)? Don't just give the answer---explain it carefully. (It is important to evaluate \(-1/2\) and \(-1\%2\) correctly. Keep in mind that in Python, when \( x \) and \( y \) are ints, we always have

\[
x = y \times (x/y) + (x\%y)
\]

where \( 0 \leq x\%y < y \).)

9. Attached to this exam are four views of your favorite clown, with many of his pixels painted white. Two of these were produced by applying the two image-processing functions shown below to the original clown image. Match the functions (func1 and func2) to the corresponding image.

```python
def func1(im):
    height=len(im)
    width = len(im[0])
    for row in range(height):
        for col in range(width):
            if row%100<50 or col%100>50:
                im[row][col]=[255,255,255]
    return im

def func2(im):
    height=len(im)
    width = len(im[0])
    for row in range(height):
        for col in range(width):
            if row%5!=0:
                im[row][col]=[255,255,255]
    return im
```