

Name: _____

Written Final Exam

This exam is closed book - you are allowed only one page of notes (double-sided). If a question seems unclear - please write down any assumptions you feel are needed. If you think that there is a just-plain mistake/typo - check with an instructor.

Anywhere we ask you what will be printed out, if you think an error will be generated, you may write “error”. You do not need to write out what the whole error message would be.

1. (250) What will print out?

```
def interp(L, i):  
    templ = "%s is at idx %d in a list with %d items"  
    vals = (L[i], i, len(L))  
    return templ % vals  
  
print interp(['you', 'are', 'a', 'genius'], 3))
```

The next few questions are based on the string `s` below, which has a much simplified set of data returned from a Twitter query.

```
s = """
[
  {
    "id": 672149962780749800,
    "user": {
      "followers_count": 823,
      "screen_name": "alimahrer",
      "id": 156844181
    }
  },
  {
    "id": 672149247811313700,
    "user": {
      "followers_count": 1569,
      "screen_name": "CatchaJob_us",
      "id": 2348426863
    }
  },
  {
    "id": 672148783413731300,
    "user": {
      "followers_count": 15575,
      "screen_name": "GymCastic",
      "id": 812521788
    }
  }
]
"""
```

2. (100) What is the type of s?

- a. Integer
- b. String
- c. List
- d. Dictionary
- e. Instance of class Tweet
- f. None of the above; that expression will generate an error

3. (100) What is the type of s.split("user")?

- a. Integer
- b. String
- c. List
- d. Dictionary
- e. Instance of class Tweet
- f. None of the above; that expression will generate an error

4. (250) What will the following code print out?

```
print len(s.split("user"))
```

5. (100) What is the type of json.loads(s)?

- a. Integer
- b. String
- c. List
- d. Dictionary
- e. Instance of class Tweet
- f. None of the above; that expression will generate an error

6. (500) Write a function **popular_tweeters** that will take a string like `s` and return a list of the screen_names in order based on their followers_count. It should pass the test at the bottom.

```
test.testEqual(popular_tweeters(s),
                ['GymCastic', 'CatchaJob_us', 'alimahrer'])
```

7. (300) What will the following code print out?

```
x = -1
y = -2
z = -3
def h(x, y = 2, z = 3):
    print x, y, z

h(x=y, z=5)
```

8. (300) What will the following code print out?

```
def hmm(L):
    start_nums = range(len(L))
    print start_nums
    nums = [x + 1 for x in start_nums]
    print nums
    return zip(L, nums)

print hmm(['a', 'b', 'c'])
```

9. (300) The code below is supposed to output the most common word in the file test.txt. When run, it gives an error message, shown below. Edit the code so that it will correctly print out the most common word.

```
1 f = open('test.txt', 'r')
2 contents = f.read()
3 d = {}
4
5 for w in contents.split():
6     d[w] += 1
7
8 dkeys = d.keys()
9 most_used = dkeys[0]
10 for k in dkeys:
11     if d[k] > d[most_used]:
12         most_used = k
13
14 print "The most used word is '"+most_used+"', which is used "+str(d[most_used])+" times"
```

```
Traceback (most recent call last):
  File "exam.py", line 6, in <module>
    d[w] += 1
KeyError: 'This'
```

The contents of the file are as follows:

```
1 This file has four lines.
2 This is the second line.
3 Line three has five words.
4 The last line has no period
```

10. (500) Define a function `join_strings` that takes two inputs, a list of strings and a separator string. It returns a single string that contains all of the original strings concatenated together, separated by the separator string.

```
def join_strings(L, sep):
```

```
    joined = join_strings(['a', 'b', 'c'], "|")
    test.assertEqual(joined, "a|b|c")
    joined2 = join_strings(['a', 'b', 'c'], "")
    test.assertEqual(joined2, "abc")
```


11. (500) What will print out?

```
def rotate_char(c, positions = 1, alphabet =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ"):
    try:
        idx = alphabet.index(c)
        rotated_idx = (idx + positions) % len(alphabet)
        return alphabet[rotated_idx]
    except:
        print "%s in not in %s" % (c, alphabet)
        return c

print rotate_char('I')
print rotate_char('Z')
print rotate_char('?')
print rotate_char('I', positions = -1)
```

12. (500) Define a function **encrypt**, that takes a string as input and returns a new string. The function should include the following:

- First use map or a list comprehension to generate a list of the results of running rotate_char on each letter in the string.
- Then run the join_strings function you defined in a previous problem to turn the list of characters into a string.
- Return that string.
- Your function should pass the tests at the bottom of the page. **Note the parameters that are passed in the calls to encrypt.**

```
test.assertEqual(encrypt("HAL", 1), "IBM")
test.assertEqual(encrypt("IBM", -1), "HAL")
```

We have provided you with a separate code printout of the revised Tamagotchi program from the textbook (with a couple small additions that will be mentioned below). For the next several questions, you'll be writing code that modifies that program.

13. (300) On lines 174-176, the program iterates through all the currently adopted pets, calling `clock_tick` and accumulating a string representation of the pet's current state. Write code to be inserted prior to line 174 that **makes the `animals` variable be a list that is sorted from least to most hungry**. That is, if there are two pets, named "Fido" and "Grand", whichever has a higher value for its hunger instance variable should be processed first.

14. (300) Generate a list of only the pets that are bored. Imagine that this code will be inserted just before line 174. Using a filter or a list comprehension, write code to generate a list of only those animals whose boredom value is above the boredom threshold for its class. (Note that on line 80, we have set a different boredom threshold for the Bird class).

15. (300) Generate a list of (boredom, hunger) tuples, one for each pet. Imagine that this code will be inserted just before line 174, so that the variable `animals` is bound to a list of instances. Use `map` or a list comprehension.

16. (500) When the user quits, the program should write a `.csv` file with information about all the adopted pets and their current states. Imagine that this code will be inserted just before line 140. For example, if there are three pets, the `.csv` file should contain the following. To make it simpler, you just have to record the name, hunger level, and boredom level, not which kind of pet they are or what sounds they've learned.

```
Name, hunger, boredom
Fido, 3, 7
Grand, 4, 2
```

17. (100) To test the teach method of the Pet class, would you use a return value test or a side effect test?
- a. Return value test
 - b. Side effect test
18. (400) Write a test for the teach method of the Pet class.

19. (500) Define a subclass of Pet called Cyborg. It works just like a regular pet, but it only starts to get hungry when it is bored. That is, when its boredom value is not above the boredom threshold, its hunger does not increment. Write the least amount of code necessary to implement the class (i.e., only write those methods that you need to.)

20. (250) You have been part of an open source project from its founding. A new leader has emerged and, well, let's just say you don't get along. The leader has convinced the rest of the core group to kick you out. The project is still remaining an open source project. Which of the following things will you still be able to do? Circle all that apply
- a. Download the latest version of the code
 - b. Run the code
 - c. Modify the code
 - d. Get your modifications incorporated into the project's github repository
 - e. Distribute a modified version, including much of the original code but with some of your modifications, through a different github repository
21. (350) Which of the following are motivators that Weber describes for people participating in open source projects? Circle all that apply.
- a. An artistic desire to write code that elegantly solves challenging problems;
 - b. Getting assigned to participate in the project as part of their paid employment;
 - c. Getting a free education by having their code critiqued and seeing how other people write good code;
 - d. A belief that no else can write code as well as they can, so their participation is essential;
 - e. A belief that socialism is a better economic system than capitalism
 - f. An ideological commitment to open rather proprietary software;
 - g. Building a reputation in order to get other benefits outside the project such as consulting gigs or job offers
22. (300) You have written a small UNIX utility program (analogous to cat or grep) for your own purposes. It's been very useful for you and you start to get requests from other people to share it. You don't mind sharing it freely, but you decide that it would bother you very much if somebody else started selling your code and keeping the money for themselves. What kind of license should you put on your code when you distribute it: GPL or BSD? Justify your answer in 2-3 sentences.