

Com S 127
Fall 2016
Topics and review problems for Final Exam
Thursday, December 15, 12:00 - 2:00 in Durham 171

General information

This will be a 120-minute, timed, pencil-and-paper written exam. No books, no notes, no electronic devices, no headphones, no collaboration. The problems will primarily involve writing Python code or reading and interpreting Python code.

Summary of Exam topics

The exam will cover everything we've done. This corresponds to the posted chapters of the textbook (*Eels in my hovercraft*), **and** *How to Think Like a Computer Scientist*, Chapters 10 and 14, which cover lists and objects, respectively. See the online topics page for exact readings associated with each topic.

Note that, by necessity, **the exam is cumulative**. Everything we have been doing recently depends on the basics covered on the first two exams. You should review Exam 1 and Exam 2, and the corresponding review sheets. A reasonable estimate is that about 40% to 50% of the exam will be on "new" material. See the review sheet from Exam 1 for general studying advice.

The list below is a rough summary of what we have covered since the last exam.

- Reading text files
- Two-dimensional lists (lists of lists)
- Nested loops
- Global variables and the `global` keyword
- Objects and classes
- Lists of objects
- Binary arithmetic
- Differences between Python and Java

There won't be anything on turtles. You should be familiar with string and list methods and how to use them, but you'll be provided with a table like the one from Exam 2 so you don't have to memorize their exact names.

A few sample problems on the new stuff

1) Write a function that, given the name of a file, counts and returns the number of words in the file. (A "word" is any text separated by whitespace.)

2) Suppose a text file is in a "comma-separated-value" format and consists of many lines of the form:

```
3/9/2001,18,17,17,18,20,22,25,30,32,32,32,33,31,28,26,26,25,22,20,20,19,18,18,18
```

The first token is a month/day/year and it is followed by 24 numbers representing temperature measurements taken each hour on that date. Write a function that, given the *name* of a file of this form, prints out each date followed by the average temperature for that date.

3) In the loop below that displays the contents of a text file, why is there an extra blank line printed in between each line of the file? Give two ways to correct the problem.

```
f = open("somefile.txt")
for line in f:
    print(line)
f.close()
```

4) Suppose you have a directory `mystuff` with two subdirectories, `cs127` and `other`, and that `cs127` is the *current working directory* for a Python program that is going to open and read a file. That file is called `my_data.txt` and is located in `mystuff`. What should you put in the parentheses to specify the file to be opened?

```
f = open( ? )
```

5) A vending machine contains a certain number of drinks. For simplicity assume there is only one kind and there is a fixed price. You interact with a vending machine by inserting some amount of money. Then you can either press the button to buy a drink or press a "coin return" button to get your money back. If you have inserted enough money for a drink, you get a drink, and its price is deducted from your balance.

Suppose we model a vending machine with a Python class `VendingMachine` having the following operations:

```
insertMoney(value)
- inserts a coin or bill with the given value
getBalance()
- returns the amount of your balance
pressCoinReturn()
- reduces your balance to zero
purchase()
- reduces your balance by the drink price and reduces by 1 the number of drinks in the machine,
provided you have put enough money in and there is a drink available to sell (otherwise does
nothing)
```

In addition, there will need to be an `__init__` method in which we can specify the drink price and the number of drinks that are initially in the machine:

```
def __init__(self, num_drinks, drink_price)
```

(Problem 5, continued)

- a) Write a set of test cases that will verify that these methods work correctly. For each call to `getBalance()`, print the actual and expected values you expect. (Be sure to include cases where `purchase()` might fail for one reason or another.)
 - b) Write a complete implementation of a `VendingMachine` class according to this specification.
- 6) For the data file of problem 2, define a type of object that can store a date and the corresponding average temperature. Then write a function to read the file return a *list* of such objects.
 - 7) Explain the difference between a *function* and a *method*.
 - 8) Explain the purpose of an *initializer*.
 - 9) Name three significant differences between Java and Python.
 - 10) Perform the binary addition:

```
  11011
+ 0111
-----
```

- 11) Write a function that creates a 2D list of numbers in which the number in row r , column c is the value $r * c$. (More or less a multiplication table.)
- 12) Write a function that, given a 2D list of numbers (a "grid"), returns a list (one-dimensional) in which the element at index i is the sum of column i of the grid.
- 13) Consider the Python script

```
x = 42

def foo():
    x = x + 1
    print(x)

foo()
```

Why does this script fail with the error: `'x' referenced before assignment` ?