Com S 127x
Fall 2015
Topics and review problems for Exam 1
Monday, October 5, in class

General information

This will be a 50-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 covers everything we have done in class through Friday, September 25 and in labs through Lab 5. This corresponds pretty well to the first 6 chapters of the textbook, although we have not discussed the third chapter (Debugging Interlude) or the last half of Chapter 6 (Functions) in detail.

Note in particular that you will not need conditionals ("if" statements) on the exam.

The list below is a rough overview of the highlights.

- expressions and statements
- variables and assignment
- calling functions
- arguments and parameters
- calling functions in a module
- problem-solving strategies
- for-loops
- ranges
- using an accumulator variable
- updating an auxiliary variable in the loop
- turtle basics
- defining functions
- local variables and scope
- functional decomposition
- reading and tracing code
How do I study?

(Excerpted from Lab 5) You can become a good at problem-solving and coding with practice and experience. Test-taking also involves some strategy. What kind of problem am I being asked to solve? Am I reading existing code, answering a question, writing a few statements, writing a function, or what? Am I printing output? Am I reading input? If I am writing a function, what are its parameters? Does it return a value, or does it perform some other action (like drawing or printing)?

Given a problem, think of a concrete example to be a "test case". That is, if you wrote the code and wanted to know whether it was correct, how would you check? Often this involves some hand calculation, and often the hand calculation shows you how to write the code.

Try to describe to yourself the steps in solving the problem. If there is a loop involved, what action is being repeated? How many times? Is it a simple repetition, or does the action change each iteration? Can I just use the loop variable, or do I need to update some auxiliary variable?

If you can’t see how to solve the problem, think about whether you can solve a related simpler problem, or solve part of the problem.

After writing the code on paper, try to read it. Does it make sense to you? Trace through it by hand and make sure it is really doing what you intended at every step.

Type up your solution in Wing or activecode, and try it using your test case.

If you weren't successful, ask yourself: how can I remember the way to solve this kind of problem? How can I recognize a similar problem in the future? Can I make up more problems that are like this one? What information can I stash in my brain to make it easier the next time?

Be the Professor!

Another excellent studying strategy is to put yourself in the professor's shoes. (It doesn't matter whether it's the red sneakers or the blue sneakers.) Imagine you are writing the exam. What would you put on it? What have we actually covered in the class? Come up with some questions that are not too easy, not too hard, that would test the skills we've practiced.

Are there answers?

No. If you have can't figure out whether your code is correct, post it on Piazza for other students, the TAs, and the prof to comment.
Problems to practice on

Be sure you can do the problems from Lab 5,
http://web.cs.iastate.edu/~smkautz/cs127f15/labs/lab5/lab5_problems.pdf

Be sure you can do the problems on the worksheet from 9/25,

Be sure you can do the problems from the quiz on September 23,

Be sure you can do the problems on the worksheet from 9/2,
http://web.cs.iastate.edu/~smkautz/cs127f15/examples/week2/problems.pdf

More problems!

1) For each snippet of code, identify the output, or if there is an error, state what is going wrong.

a)
```python
a = 42
b = a
a = a + 1
print(b)
```

b)
```python
x = 42
x + 1
print(x)
```

c)
```python
def foo(x):
    x = x + 1
    print(x)
x = 42
foo(x)
print(x)
```

d)
```python
def area(x, y):
    result = x * y
    return result
x = 10
y = 20
area()```
e) def mystery(p):
    for count in range(5):
        y = p % 5
        p = p - y - 1
    print(p)
mystery(20)

2) What's the difference between an argument and a parameter for a function? (Explain briefly and give an example of each.)

3) Explain why in Python, y = x doesn't mean the same thing as x = y.

4) For each expression in the table below: if the expression is a valid Python expression give its value and its type (int, float, boolean, string); if not a valid expression, put “invalid” in both columns. Assume that the following variables have already been initialized as shown:

   i = 25
   s = "boogers"
   x = 3.2

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>i // 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>i / 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>i &lt; 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pow(len(s), 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>i + x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>i % 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x / 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s + s</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 = i</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5) Write an interactive program that prompts the user to enter a number of eggs, and then prints out how many dozens there are and how many are left over.

6) a) Write a range expression for the list [7, 9, 11, 13, 15]
   
b) Write a range expression for the list [100, 95, 90, ..., 0]

7) Write a function that, given a number n, returns the sum of the first n terms in:
   1 + 1/2 + 1/4 + 1/8 + ...  

8) Write a function that, given n and k, returns the product of the k numbers
   n(n - 1)(n - 2)...(n - k + 1)
9) The value in an account goes up by some percentage each year. Write a function that, given the initial balance, the percentage, and a number of years, prints out the value in the account for each year.

10) For each of problems 5 through 9 on this review sheet, make up different problem that tests the same concepts or skills!

11) In lab we created a function for drawing a polygon, given a turtle, the length of each side, and the number of sides. Here is an example of the code:

```python
def draw_polygon(t, size, n):
    angle = 180 * (n - 2) / n
    turn_angle = 180 - angle
    for count in range(n):
        t.forward(size)
        t.left(turn_angle)
```

For n = 7, the result is similar to what is pictured here:

![Octagon](image)

Write a function that, given an odd number n, draws lines through alternate corners to create a star like this:

![Star](image)

(You do not have to worry about centering the drawing. Assume that the 'size' parameter is the length from corner to corner of the star.)