CompSci 101
Introduction to Computer Science

March 31, 2015

Review for exam

Prof. Rodger

Thanks to Elizabeth Dowd for giving this lecture

Announcements

- Exam 2 is Thursday
- Assignment 7 is out
- APT 8 is due today
- Review Session with Prof. Rodger
  - Wed. 5:30pm, LSRC B101
- Finish lecture notes from last time

Snarky Hangman

- Version of Hangman that is hard to win.
- Program keeps changing secret word to make it hard to guess!
- User never knows!
- Once a letter is chosen and shown in a location, program picks from words that only have that letter in that location
- Program smart to pick from largest group of words available

Snarky Hangman - Dictionary

- Builds a dictionary of categories
- Start with list of words of correct size
- Repeat
  - User picks a letter
  - Make dictionary of categories based on letter
  - New list of words is largest category
    - Matched letters
    - Letters guessed by not chosen
    - List shrinks in size each time
Snarky Hangman Example

- Possible scenario after several rounds

(secret word: lucky) # words possible: 33
Progress: _u___
letters missed: a be i o s
guess a letter: c

- You currently have a list of all words with u the second letter. From that build a dictionary of list of words with no c and with c in different places (show count of number of words in each list):

<table>
<thead>
<tr>
<th></th>
<th>Only 2 words of this type</th>
<th>Only 2 words of this type</th>
<th>Only 8 words of this type</th>
</tr>
</thead>
<tbody>
<tr>
<td>cu___ 2</td>
<td>uo___ 2</td>
<td><em>u_c</em>__ 2</td>
<td><em>u</em>__ 21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Choose “no c”, most words, 21

Exam logistics

- Only need a pen or pencil
- No scratch paper
- See the reference sheet of Python information you will get with the test (see resources page)
- Closed book, closed notes, closed neighbor
- Covers lecture, lab and assigned reading
- Have put old quizzes back up as quiz review – This is NOT for a grade, for studying only

Understand old and new topics

- Old topics: if, for, while, lists, strings
- list comprehension, enumerate
- Files – write code - Will give you a file already opened and ready for reading
- Sets, Dictionaries – write code – create and use
- Understand items on Python review sheet on resources page
- HAVE NOT COVERED TOPICS – regular expressions or recursion

The best way to study

- Write code on paper!
- Resources page has old tests and solutions – Try writing code, then look at solutions
- Rewrite an APT
- Rewrite code we did in lecture
- Rewrite code we did in classwork or lab
Think about how parameters work

- Look at variables before passed to a method
- What happens to them after they are passed?
- Can the parameter and argument have the same name?

```python
def testone(num, s):
    num = 4
    s = "Hello"
    print 'In testone, num is ', num, ', s is ', s

'example 1'
print 'example 1'
number = 5
school = 'duke'
print 'number is ', number, ', school is ', school
testone(number,school)
print 'After call to testone, number is ', number, ', school is ', school

'example 2'
def testone(num,s):
    num = 4
    s = "Hello"
    print 'In testone, num is ', num, ', s is ', s

'example 2'
def testone(num,s):
    num = 2
    s = 'cat'
    print 'num is ', num, ', s is ', s
testone(num,s)
print 'After call to testone, num is ', num, ', s is ', s
```

Parameter and Argument have the same name

```python
def testtwo(lista):
    lista.append(6)
    print 'In testtwo, lista is ', lista

'example 3'
def testtwo(lista):
    lista.append(6)
    print 'In testtwo, lista is ', lista

'example 3'
def testtwo(lista):
    lista.append(6)
    print 'In testtwo, lista is ', lista
```
Parameter and Argument have the same name

```python
def testtwo(lista):
    lista.append(6)
    print 'In testtwo, lista is ', lista

''' example 4'''
print 'example 4'
lista = [4,3,8]
lista.append(5)
print 'lista is ', lista
testtwo(lista)
print 'After call to testtwo, lista is ', lista
```

```python
def testthree(lista):
    print 'In testthree, lista is ', lista
    lista = (7,1,3)
    print 'In testthree, lista is ', lista

''' example 5'''
print 'example 5'
lista = (4, 5, 2)
print 'lista is ', lista
testthree(lista)
print 'after call to testthree, lista is ', lista
```

Parameter and Argument have the same name

```python
def testfour(lista):
    print 'In testfour, lista is ', lista
    lista[1] = (4,3)
    print 'In testfour, lista is ', lista

''' example 6'''
print 'example 6'
lista = [(4, 5, 2),(3,2),(3,4)]
print 'Lista is ', lista
testfour(lista)
print 'after call to testfour, lista is ', lista
```

```python
def testfour(lista):
    print 'In testfour, lista is ', lista
    lista[1] = (4,3)
    print 'In testfour, lista is ', lista

''' example 7'''
print 'example 7'
mylist = [(4, 5, 2),(3,2),(3,4)]
print 'mylist is ', mylist
testfour(mylist)
print 'after call to testfour, mylist is ', mylist
```
Now go over Test Practice problems

```python
def testfive(seta):
    print 'In testfive, seta is ', seta
    seta.add(5)
    seta.add(3)
    print 'In testfive, seta is ', seta

'example 8'
print 'example 8'
myset = set([4, 5, 2])
print 'myset is ', myset
testfive(myset)
print 'after call to testfive, myset is ', myset

def testsix(seta):
    print 'In testsix, seta is ', seta
    seta.add((5,8))
    seta.add((3,2))
    print 'In testsix, seta is ', seta

'example 9'
print 'example 9'
myset = set([(4,3), (5,2,1), (3,2)])
print 'myset is ', myset
testsix(myset)
print 'after call to testsix, myset is ', myset
```