

CompSci 101

Introduction to Computer Science



Sept 12, 2017
Prof. Rodger

compsci 101, fall 2017

1

Announcements

- Reading and RQ5 due next time
- Assignment 2 due Thursday
- APT 1 is due today, APT 2 out today
- Catch up Schedule on main web page
- Lab 3 – splicing, making decisions
- Today – TWOTS
 - Solving problems – 7 Step process
 - Decisions - if, Boolean

compsci 101, fall 2017

2

Assignment 2

- Questions?



compsci 101, fall 2017

3

Submitting Assignment 2

- Use Ambient/eclipse to submit!
 - Check if submitted with Submit History – files submitted should be listed!
 - Alternative submit – use websubmit – on assign tab
 - What time is it due? Thursday 11:59pm

compsci 101, fall 2017

4

Why is this person so important
to this course?



compsci 101, fall 2017

5

Why is this person so important
to this course?



- Have you donated yet?

compsci 101, fall 2017

6

Top 10 list for surviving in CompSci 101

10. Read the book and Ask questions
9. Eat lots of pizza
8. Learn how to spell Rodger
7. Understand what you turn in
6. Follow the 7 step process

compsci 101, fall 2017

7

Top 10 list (cont)

5. Check Piazza every day
4. Visit your prof in her office
3. Learn how to debug your programs
2. Seek help (one hour rule!)
1. Start programming assignments early

compsci 101, fall 2017

8

Finish from last time.....

Function Detective

- <http://bit.ly/101f17-0907-5>

```
if __name__ == "__main__":  
    f = open("words.txt")  
    for w in f:  
        w = w.strip()  
        print w, pluralize(w)
```

Finish from last time.....

Function Detective

- <http://bit.ly/101f17-0907-5>

```
if __name__ == "__main__":  
    f = open("words.txt")           # f is the file  
    for w in f:                     # for each line in the file  
        w = w.strip()              # remove carriage return from line  
        print w, pluralize(w)      # process word
```

in this example, each line happened to be just one word,
this loop iterates over lines

Another way for the main

Iterate over words, not lines

```
if __name__ == "__main__":  
    f = open("words.txt")  
    all = f.read()  
    wordlist = all.split()  
    for w in wordlist:  
        print w, pluralize(w)
```

Another way for the main

Iterate over words, not lines

```
if __name__ == "__main__":  
    f = open("words.txt")           # f is the file  
    all = f.read()                  # all is the file as one string,  
                                    # carriage returns are removed!  
    wordlist = all.split()          # split string into list of words  
    for w in wordlist:              # iterate over the words  
        print w, pluralize(w)       # process each word
```

would work if there were multiple words on a line in the file

Python – Names and Types

- Names vs abstractions
 - What is <http://152.3.140.1>
 - What is <http://www.amazon.com>
- Types are important
 - What is foo.pdf, foo.mp4, foo.jpg, foo.wav
 - Do the file extensions guarantee file type?

- Python – what types are these?

```
first = "Susan"
```

```
x = 6
```

```
y = 3.4
```

compsci 101, fall 2017

13

Strings

- Sequence of characters in quotes
 - "I" + 'Love' + '''Python'''
 - "I" 'Love' '''Python'''
- String operators: concatenation (+), repeat(*)

- Precedence?

```
"a" + "b" + "c" * 3
```

- Precedence?

```
"a" + "b" "c" * 3
```

compsci 101, fall 2017

14

Strings

- Sequence of characters in quotes (same result)

```
"I" + 'Love' + '''Python'''
```

```
"I" 'Love' '''Python'''
```

```
'ILovePython'
```

- String operators: concatenation (+), repeat(*)

- Precedence?

```
"a" + "b" + "c" * 3
```

```
'abccc'
```

- Precedence?

```
"a" + "b" "c" * 3
```

```
'abcbcbcb'
```

compsci 101, fall 2017

15

Names, Types and Values

- bit.ly/101f17-0912-1

compsci 101, fall 2017

16

Names, Types and Values

```
import urllib2
if __name__ == "__main__":
    source = urllib2.urlopen("http://.../poe.txt")
    s = source.read()
    words = s.split()
    total = len(s)
    all = len(words)
    print total > all
```

compsci 101, fall 2017

17

Names, Types and Values

```
import urllib2
if __name__ == "__main__":
    source = urllib2.urlopen("http://.../poe.txt")
    s = source.read()
    words = s.split()
    total = len(s)
    all = len(words)
    print total > all
```

source is the file
s is the file as one long string
words is list of words from s
total is the number of char in s
all is the number of words in s
True, more char than words!

compsci 101, fall 2017

18

Grace Murray Hopper (1906-1992)

- “third programmer on world's first large-scale digital computer”
 - US Navy: Admiral

“It's better to show that something can be done and apologize for not asking permission, than to try to persuade the powers that be at the beginning”



<https://www.youtube.com/watch?v=1-vcErOPofQ>

□ ACM Hopper award given for contributions before 35

2010: Craig Gentry: <http://www.youtube.com/watch?v=qe-zmHoPW30>

2011: Luis von Ahn

2013: Pedro Felzenszwalb

2014: Sylvia Ratnasamy

2015: Brent Waters



APT: Pancakes

Problem Statement

You're a short-order cook in a pancake restaurant, so you need to cook pancakes as fast as possible. You have one pan that can fit *capacity* pancakes at a time. Using this pan you must cook *numCakes* pancakes. Each pancake must be cooked for five minutes on each side, and once a pancake starts cooking on a side it has to cook for five minutes on that side.

However, you can take a pancake out of the pan when you're ready to flip it after five minutes and put it back in the pan later to cook it on the other side.

Write the method, *minutesNeeded*, that returns the shortest time needed to cook *numCakes* pancakes in a pan that holds *capacity* pancakes at once. See the examples.

Specification

filename: Pancakes.py

```
def minutesNeeded (numCakes, capacity):
    """
    return integer representing time to cook pancakes
    based on integer parameters as described below
    """
```

compsci 101, fall 2017

20

Examples

1. `numCakes = 0`
`capacity = 4`

Returns: 0

It takes no time to cook 0 pancakes.

2. `numCakes = 2`
`capacity = 2`

Returns: 10

You cook both pancakes on one side for five minutes, then flip them over and cook each on the other side for another five minutes.

compsci 101, fall 2017

21

APT Pancake:

- How do you solve this problem?
 - First steps: are there simple cases that can be solved immediately?
 - What are these for the pancake problem?
 - Sometimes it helps to know if you are on track, should you use Python to check your paper and pencil work?
- Get specific, solve for 5, not N
 - Fix one parameter, vary the other
 - Identify the cases and continue



compsci 101, fall 2017

22

Solve an APT - Pancakes

bit.ly/101f17-0912-2



compsci 101, fall 2017

23

Problem Solving to Code

7 Step Process

1. Work small examples by hand
2. Write down what you did in words (algorithm)
3. Find Patterns (generalize algorithm)
4. Work another example by hand (does your algorithm work? If not, go back to 2)
5. Translate to code
6. Test several cases
7. Debug **failed** test cases

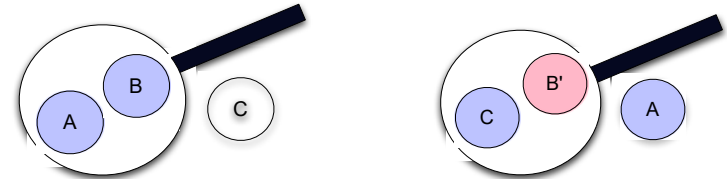
24

Pancake Problem

- Work through the 7 step process....

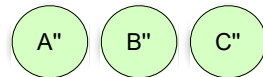
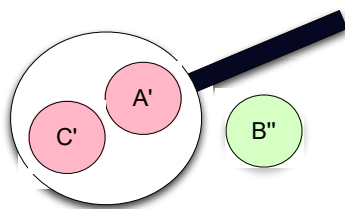
Three pancakes in a two-cake pan...

- Number of cakes in the system
 - First 5 minutes
- Number of cakes in the system
 - Second 5 minutes



Three pancakes in a two-cake pan...

- Number of cakes in the system
 - Third 5 minutes
- How many minutes to cook all three pancakes?



How to solve problems with different cases?

- Keep score in a video game?
 - Different points for different tasks?
- Translate a book from English to Spanish?
 - Different words, different rules
- Identify proteins in strands of DNA?
 - Start codon: atg Stop Codon: tag
- Different cases with Pancake APT?
- In Python use: if, else, elif

How to teach pancake Flipping

- http://www.youtube.com/watch?v=W_gxLKSsSIE
 - Is this computer science? <http://bit.ly/zykOrh>
 - For longer, more complex robotic tasks
 - <http://www.youtube.com/watch?v=4usoE981e7I>



compsci 101, fall 2017

29

Review Functions

www.bit.ly/101f17-0912-3

```
def duplicate(word, num):  
    answer = word * num  
    return answer
```

```
def duplicate2(word, num):  
    answer = word * num  
    print answer
```

```
def duplicate3(word, num):  
    answer = word * num
```

```
1. print duplicate ("Go", 3)  
2. print duplicate2("Go", 5)  
3. print duplicate3("Go", 2)  
4. duplicate("Go", 5)  
5. duplicate2("Go", 4)  
6. duplicate3("Go", 2)
```

compsci 101, fall 2017

30