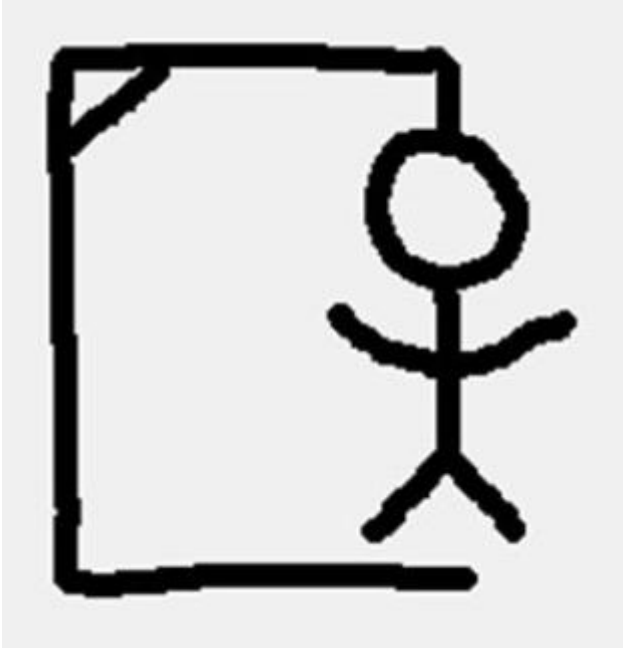
CompSci 101
Introduction to Computer Science



October 16, 2014

Prof. Rodger

Announcements

- Reading for next time on calendar page
  - RQ 10
- Assignment 5 out today – Hangman –
  - due in a week
- APT 5 is due on Tuesday
  - Note added one APT – EatingGood – try first

- Finish lecture notes on sets from last time
- Focus on Debugging – trace through on paper!

Debugging Problems

- Today the focus is on debugging.
- There are several problems. Trace by hand to see if you can figure out if they are correct or not, or what to do to correct them.

Found in Mark II
(9/9/1947)

Links to forms

Problems 1-3:

www.bit.ly/101fall14-1016-01

Problems 4-5:

www.bit.ly/101fall14-1016-02

Problem 1 – Does it work?

- The function *sizes* has a parameter named *words* that is a list of strings. This function returns a list of the sizes of each string. For example, sizes(['This', 'is', 'a', 'test']) should return the list [4, 2, 1, 4]

```
def sizes(words):
    nums = [ ]
    for w in words:
        nums = len(w)
    return nums
```

Problem 2 – Does it work?

- The function *buildword* has a parameter *words* that is a list of strings. This function returns a string that is made up of the first character from each word in the list. For example, buildword(['This', 'is', 'a', 'test']) returns 'Tiat'

```
def buildword(words):
    answer = ''
    for w in words:
        answer += w[:1]
        return answer
```

Problem 3 – Does it work?

- The function *middle* has a parameter *names* that is a list of strings, which each string is in the format "firstname:middlename:lastname". This function returns a list of strings of the middlenames.

```
def middle(names):
    middlelist = []
    for name in names:
        name.split(":")
        middlelist.append(name[1])
    return middlelist
```

Problem 4 – Does it work?

- The function *removeOs* has one string parameter named *names*. This function returns a string equal to names but with all the lowercase o's removed.

```
def removeOs(word):
    position = word.find("o")
    while position != -1:
        word = word[:position] +
            word[position+1:]
    return word
```

Problem 5 – Does it work?

- The function uniqueDigits has one int parameter number. This function returns the number of unique digits in number. If the number is 456655, then it returns 3.

```
def uniqueDigits(number)
    digits = [ ]
    while number > 0:
        digits.append(number % 10)
        number = number / 10
    return len(digits)
```