Introduction to:
Computers & Programming:
Post-Midterm 1 Review

Adam Meyers
New York University
Summary

• Some Procedural Matters
• Review of Variables, Functions and Loops
• Review of Part 1
• Review of Part 2
• Lab Tomorrow
## Grading Curve

<table>
<thead>
<tr>
<th>Letter Grade</th>
<th>Raw Score 007</th>
<th>Raw Score 011</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>87 and Above</td>
<td>90 and Above</td>
</tr>
<tr>
<td>A-</td>
<td>80-86</td>
<td>83-89</td>
</tr>
<tr>
<td>B+</td>
<td>68-79</td>
<td>72-83</td>
</tr>
<tr>
<td>B</td>
<td>60-67</td>
<td>64-71</td>
</tr>
<tr>
<td>B-</td>
<td>52-59</td>
<td>56-63</td>
</tr>
<tr>
<td>C+</td>
<td>46-51</td>
<td>50-55</td>
</tr>
<tr>
<td>C</td>
<td>41-46</td>
<td>41-49</td>
</tr>
<tr>
<td>D</td>
<td>25-40</td>
<td>25-40</td>
</tr>
<tr>
<td>F</td>
<td>24 and Below</td>
<td>24 and Below</td>
</tr>
</tbody>
</table>
Raw Score vs Letter Score

• Only the Letter Score counts for purposes of the final grade, e.g., there is no difference between a 72 and a 75 if both are part of the same range

• It is only worth haggling over your grade if:
  – the change in score will effect your letter grade, e.g.,
    • raw score is 89 (an A-) and discrepency is worth 1 point
    • discrepency is worth 5-10 points
    • Etc.

• Of course, understanding everything you got wrong is important regardless of the grade.
Grading Considerations

• If Midterm 2 is significantly better than Midterm 1, Midterm 2 will count more towards the final grade than Midterm 1.
  – If it is a real outlier, I will not count it at all, e.g.,

  • If (1st Midterm == D) and (2nd Midterm == A) and (Final == A) and (Homework Grade == A):
    Final Grade = A

• In other words improvement over the course of the class can have a big influence on final grade. I have not figured out exactly how to quantify this, but it will be a major factor.

• Main purpose of final grade: An indicator of state of knowledge at the end of the class.

• Administrative purpose: Only allow qualified students to progress to higher level CS classes. C is the minimum grade in this class if you want to take the JAVA programming class.
Timing Issues

• I will look for ways to shorten next midterm
  – Part 1:
    • fewer or shorter questions
    • choice of 4 out of 5
  – Part 2:
    • more pre-written functions for use
    • more algorithmic guidance

• Students:
  – more practice programming
  – try to extend lab problems more
Variable Assignment

- **Correct:** Name_of_Variable = Value
- **Incorrect:** Value = Name_of_Variable
- **Correct:**
  - `def function1():
    ## all sorts of stuff
    return(5)
  
  - `def function1():
    ## all sorts of stuff
    variable1 = function1()
    ## variable1 is set to whatever function1 returns

- **Incorrect:**
  `def function1 ():
    variable1 = 5
  
def function2 ():
    blah blah blah
    function1()
    variable2 = variable1    ### ERROR
    ## variable1 is “local” to function1—variable1 doesn't exist in function2`
Variable Assignment 2

• You can use global variables, but you need to declare them as global

• \texttt{variable1} = 5

  \begin{verbatim}
  def function1():
      global variable1
      variable2 = variable1
  \end{verbatim}

• It is usually better to use local variables

• Global variables are only appropriate when all or most of the functions in your (large) program use/update them.
Function Declaration

• Functions can call other functions
  
  ```python
def function1(blah):
    return(blah+blah)
def function2(blah):
    return(blah*2)
def function3(bing):
    return(function1(bing)+function2(bing))
  ```

• But do not define functions inside of each other
  
  ```python
def function1(blah):
    def function2(blah):
      ....
  ```
How Loops Work with Accumulating Variables

• def intersperse (string1,string2):
    big_string = " ## initialize big_string as empty
    if len(string1)==len(string2):
        for index in range(len(string1)):
            big_string=big_string+string1[index]+string2[index]
        print(big_string)
    return(big_string)

• def triangle_number(number):
    triangle_output = 0
    for num in range(1,number+1):
        triangle_output = triangle_output+num
        print(num,triangle_output)
    return(triangle_output)
How Nested Loops Work

• def print_nesting(string1,string2):
  big_string = "
  for char1 in string1:
    print('loop for',char1)
    for char2 in string2:
      print('loop for', char2)
      big_string = big_string+char1+char2
    print(char1,char2,big_string)
  return(big_string)
The Tests in PDF form

• 007 test:
  – http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-007/midterm1-version2.pdf

• 011 test:

• Makeup test:
  – http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-007/midterm1-version1a.pdf
Part 1 of Test

• 007 test:
  – http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-007/midterm1-version2.py

• 011 test:
  – http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-007/midterm1-version1.py

• Makeup test:
  – http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-007/midterm1-version1a.py
Intro to: Computers & Programming
Post-Midterm 1 Review
CSCI-UA.0002

Section 007 & Makeup
Part 2: Question 6

• Input is a sentence, which is a string.
• The output should be a string of the same length, but with 'C','V','A' or 'O', rather than the original characters.
• The program should loop through the string one character at a time, choosing one character to be included in the output string. There are two ways to do this correctly:
  – Have an output string that starts as the empty string and accumulates instances of 'C', 'V', 'A' or 'O' and then returns that output string.
  – Print out one character at a time, with sep and end set to the empty string (this is the less preferred option, but I accepted it)
• The if and elif clauses should include:
  – If char in 'bcdfghijklmnpqrstvwxz'    ## to identify consonants
  – If char in 'aeiou'                                  ## to identify vowels
  – If char == 'y'                                       ## to identify 'y'
  – All other characters (not just spaces) should be classified as other ('O')
  – The 'in' operator can be substituted with instances of “or', e.g.,
    • If (char == 'a') or (char == 'e') or (char == 'I') or (char == 'o') or (char == 'u')
Section 007
Part 2 : Question 7

- Nobody really did this one at all.
- The way to do it ignoring leap years would be as follows:
  - Write a function converting days of the week to integers from 0 to 6
  - Write a function converting 0 to 6 to days of the week
  - num_answer = ((Num_years * 365) + current_day) % 7
  - return(convert_to_weekday(num_answer))
- For every four years in the future add 1 to num_answer.
- The information about which month it is does not seem to matter.
This is the game of craps. The rules were confusing for some.

A roll of the dice equals the sum of two instances of roll_die, e.g., dice_roll = roll_die() + roll_die.

The first dice_roll will end the game if it is a 7, 11, 2 or 12. Otherwise it should be saved for the rest of the game. In craps, the saved roll is typically referred to as “The Mark.”

If the first roll does not end the game, it is necessary to keep re-rolling until the game ends. This is best achieved with a while loop with some way of exiting should the game end.

The new_roll (e.g., new_roll = roll_die()+roll_die()) ends the game if:

- The player loses – a 7 or 11 is rolled
- The player wins – a roll matching the mark is rolled, e.g., new_roll==dice_roll)

If the game does not end, the new roll is re-rolled.
Part 2: Question 6

- This program just needs to print 5 lines given one input sentence.
- Don't allow too-long sentences, e.g.,
  - if len(sentence)>30:
    - print('Bad Sentence')
    - return(False)
- The first 2 and last 2 lines that just involve printing 60 asterisks, e.g., print(65*'*')
- Middle line combines: 15 asterisks, centered sentence, 15 asterisks
  - If combined with '+', no problem, if using 'print', remember to use sep="
  - padding = 30-len(sentence)
  - There should be spaces of length padding before and after sentence, e.g., padding//2 *' ' 
    - a number of spaces, not a number
  - Given an odd length sentence, you should also include an extra space on one part of the padding or the other.
Section 011 and Makeup
Part 2
Question 7

- Calculating Dog Years based on if statements & a table.
- 1 earth year → 14 dog years
- Other dog years require combining different multiples of years, e.g.,
  - 30 earth years = 14 + (6.5 * 4) + (5 * 5) + (10 * 4) + (30 -20) = 115
  - Broken down many different ways in student answers
  - Typically,
    - If (year > X) and (year <Y):
      \[
      \text{output} = \text{Base\_Value} + ((X\text{-Base\_Age})\text{\_Increment})
      \]
    - Example:
      - if (year> 10) and (year <20):
        \[
        \text{output} = 65+((\text{year}-10)*4)
        \]
      - Do to context of ifs and elifs (year>X) is often unnecessary
The player picks a number between 1 and 1000

There should be a loop (for or while) keeping track of the computer's guesses.

- The loop should exit if the program guesses correctly
  - use return or break
- The loop should exit after 10 unsuccessful guesses
  - use return, break or a for-loop that only allows 10 guesses

Computer should initialize low to 1 and high to 1000

- Each guess is the rounded average of high and low
- User's answers:
  - “correct” will end the game
  - “high” will change high to guess-1
  - “low” will change low to guess+1
- Each change to high or low will result in a new guess