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

Adam Meyers
New York University
Summary

• Some Procedural Matters
• Select 009 Test Problems
• Select 004 Test Problems
# Grading Curve

<table>
<thead>
<tr>
<th>Raw Score</th>
<th>Letter Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>93 and above</td>
<td>A</td>
</tr>
<tr>
<td>88 and above</td>
<td>A-</td>
</tr>
<tr>
<td>84 and above</td>
<td>B+</td>
</tr>
<tr>
<td>78 and above</td>
<td>B</td>
</tr>
<tr>
<td>74 and above</td>
<td>B-</td>
</tr>
<tr>
<td>70 and above</td>
<td>C+</td>
</tr>
<tr>
<td>66 and above</td>
<td>C</td>
</tr>
<tr>
<td>62 and above</td>
<td>C-</td>
</tr>
<tr>
<td>55 and above</td>
<td>D</td>
</tr>
<tr>
<td>Below 55</td>
<td>F</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 71 and a 73 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 91 (an A-) and discrepancy is worth 2 points
    • discrepancy is worth 5-10 points
    • Etc.

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

- Final Grade = (0.05 * quizzes) + (0.2 * HW) + (0.2 * Mid1) + (0.2 * Mid2) + (0.35 * Final)
  - Each of these numbers are letter grades from F to A (i.e., 0 to 4)
- If you missed 1 midterm:
  - Final Grade = (0.05 * quizzes) + (0.2 * HW) + (0.3 * Mid2) + (0.45 * Final)
  - If you scored better on your second midterm than the first, I will handle it this way also.
- Basically final grades these average numbers translate into the corresponding letter grades (1=D, 2=C, 2.7=B-, 3.0=B, 3.3=B+, 3.7=A-, 4.0=A)
  - Numbers in between are rounded up or down
  - Rounding preferences: Final Exam; did you do HWs?; Class Participation, ...

Main purpose of final grade: Indicator of state of knowledge at end of class.
Administrative purpose: Allow qualified students to progress to higher level CS classes. C is the minimum if you want to take the JAVA programming class.
The Tests in PDF form
Answers as PY files

• 004 test:
  – http://cs.nyu.edu/courses/spring16/CSCI-UA.0002-004/midterm2_version2.py

• 009 test:
  – http://cs.nyu.edu/courses/spring16/CSCI-UA.0002-004/midterm2_version1.py
009 Test: Question 4 – Roman Numerals

• Figure out what program does, not what a well-formed roman numeral is.
• total – accumulator variable
• Processes previous_number based on number
  – If previous_number == 'START'
    • Set previous_number to number
  – Elif previous_number < number
    • Add in (number – previous_number) to total
    • set previous_number to 'START' (for next iteration)
  – Else
    • add previous_number into total
    • Set previous_number to number
• If previous_number does not equal 'START' when loop is complete
  – Add previous_number into total
009 Test: Question 6
(most solved the wrong problem)

- Sample input:
  - react_times = [[1,2,3,4,5],[5,2,3,2,2],[1,1,1,5,2],[3,3,3,1,1]]

- Sample output:
  - [2.5, 2.0, 2.5, 1.875, 2.5]

- react_times is a list of records
  - Each record (rows) represents an experimental subject
  - Each position in each record represents a test
  - 2.5 = average of all scores at index 0
  - 2.0 = average of all scores at index 1
  - Etc.

- A solution:
  - Initialize total as list of number of zeros as in the first record ([0,0,0,0,0] for this example)
  - Loop through list of records, totaling the values for each corresponding test
  - At the end of the loop the totals for this example: [10.0,8.0,10.0,7.5,10.0]
  - Finally, divide each of these numbers by the number of records: [2.5, 2.0, 2.5, 1.875, 2.5]
004 Test: Question 5 – ZigZag

• Step 1: orient the turtle upwards to make 1st line

• Remaining Steps
  – Go straight and alternatively turn left and right the correct number of times
  – each turn “undo” previous & turn in other direction

• Solution 1: angle = 1st turn, 2 * angle = other turns

• Solution 2: (180-angle)/2 = 1st turn, 180-angle = other turns

• I accepted both, even though the zigzags were different
  – Solution 2 most closely follows the specs of the question
  – The turns are from the “turtle's point of view”, not the angles between the lines that are drawn.
004: Question 6 – Find Name on Map

• Solve the problem in general – do not hard code a solution for the example

• Input is a list of streets
  – Each street is a list of buildings
    • Each building is at index corresponding to its avenue

• Loop through the map one street at a time
  – If you find the building on that street
    • Street = index of street + 1
    • Avenue = index of building in street + 1
  – If the building was not found after a complete search
    • It is not on the map