Introduction to: Computers & Programming: Post-Midterm 2 Review

Adam Meyers
New York University
Summary

• Some Procedural Matters
• Review of concepts that seemed to cause the most difficulty
• Review of Part 1
• Review of Part 2
## 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>87 and above</td>
<td>A-</td>
</tr>
<tr>
<td>83 and above</td>
<td>B+</td>
</tr>
<tr>
<td>78 and above</td>
<td>B</td>
</tr>
<tr>
<td>75 and above</td>
<td>B-</td>
</tr>
<tr>
<td>72 and above</td>
<td>C+</td>
</tr>
<tr>
<td>67 and above</td>
<td>C</td>
</tr>
<tr>
<td>60 and above</td>
<td>C-</td>
</tr>
<tr>
<td>40 and above</td>
<td>D</td>
</tr>
<tr>
<td>Below 40</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 83 and a 86 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 93 (an A-) and discrepancy is worth 1 point
    • discrepancy 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

• Similarly, if you do significantly better on the Final than Midterm 2, I will take that into account as well.

• 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.
How Loops Work with Accumulating Variables

- Accumulator variables
  - Initialized before loop
  - Updated During loop
  - Usually the same type after the loop as when it was initialized

- Types of accumulators
  - Lists:
    - Typically initialized as the empty list []
    - Updated with list.append(add_on) or similar
  - Numbers (integers or floats)
    - Typically initialized to 0 or 1, but can be initialized to any number
    - Updated by some mathematical operation (+, *, etc.)
  - Strings
    - Typically initialized as the empty string ("")
    - Typically updated by appending (+)
Nested Loops on the Test

- Lists of Lists = lists of records
- Each record was a list of values
- for record in list_of_lists:
  - initialize 1 or more accumulator variables
  - do something to each record
    - Calculate attribute (using accumulators) or
    - print something out or
    - change each record (lists are mutable)
- Doing something to a record may involve a (nested) loop going through some or all items in record
Turtle Graphics

• There were multiple solutions to the turtle problem in part 2 – some of them were much simpler than others

• I gave some incorrect instructions during the test and did not take off for problems related to that instruction. This is the correct characterization of turtle.circle(radius):
  – When a turtle is facing rightwards towards positive X, it draws a circle above it
  – When a turtle is facing leftwards towards negative X, it draws a circle below it
The Tests in PDF form

• 007 test:

• 011 test:
  – http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-007/midterm2-version2.pdf
Answers to Part 1
Sample Answers to Part 2

• 007 test:
  – http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-007/midterm2-version1.py

• 011 test:
  – http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-007/midterm2-version1.py