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

Adam Meyers
New York University
Summary

• Some Procedural Matters
• Review of Concepts from Test
• Review of Part 1
• Review of Part 2
• Lab Tomorrow
# Grading Curve

<table>
<thead>
<tr>
<th>Letter Grade and 0-4 Equivalent</th>
<th>Raw Score Version 1</th>
<th>Raw Score Version 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4.0</td>
<td>93 and Above</td>
</tr>
<tr>
<td>A-</td>
<td>3.7</td>
<td>90-93</td>
</tr>
<tr>
<td>B+</td>
<td>3.3</td>
<td>85-89</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>81-84</td>
</tr>
<tr>
<td>B-</td>
<td>2.7</td>
<td>76-80</td>
</tr>
<tr>
<td>C+</td>
<td>2.3</td>
<td>72-76</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>67-71</td>
</tr>
<tr>
<td>C-</td>
<td>1.7</td>
<td>60-66</td>
</tr>
<tr>
<td>D+</td>
<td>1.3</td>
<td>45-59</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>21-53</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>20 and Below</td>
</tr>
</tbody>
</table>
GradeScope

• There was a learning curve, but overall, I think it was a better way to grade.
• One change for the future: each question in section 2 will have a designated 2 or 3 pages.
• I think the feedback and organization is more consistent than the on-paper version
• There may be some scanning errors if anyone used light pencil.
• Please let me know if you think any part of GradeScope added to the normal type of grading errors.
Raw Score vs Letter Score

• Only the Letter Score counts for the final grade, e.g., there is no difference between a 72 and a 74 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 on the borderline between 2 grades
    • discrepancy is worth 5-10 points
    • Etc.

• Of course, understanding everything you got wrong is important regardless of the grade.
Your Final Grade in Python

- def calc_grade1(Quiz, HW, Mid1, Mid2, Final):
  
  # Quiz is the average grade for quizzes
  # HW is the average grade for programming assignments
  # Mid1 is your Midterm 1 grade
  # Mid2 is your Midterm 2 grade
  # Final is your grade on the Final Exam

  Grade = (Quiz * .05) + (HW * .2) + (Midterm1 * .20) + (Midterm2 * .20) + (Final * .35)
  
  return(Grade)

- def calc_grade2(Quiz, HW, Mid1,Mid2,Final):

  Grade = (Quiz *. 05) + (HW *.2) + (max(Midterm1,Midterm2) * .30) + (Final * .45)

  return(Grade)

- def final_grade(Quiz, HW, Mid1,Mid2,Final):

  Grade = max(calc_grade1(Quiz,HW,Mid1,Mid2,Final),calc_grade2(Quiz,HW,Mid1,MId2,Final))

  return(Grade)

- Bonus for A Grade on 2 Midterms: One A counts as an A+ for purpose of calculating average
Grading Considerations

• Therefore:
  – If you do better on the 2nd midterm than the first, only the 2nd midterm grade will probably count
  – If you get As on both midterms, you get a bonus of .3 added to one midterm

• Thus improvement over the course of the class can influence the final grade.

• **Main purpose of final grade**: Indicator of state of knowledge at end of class.

• **Administrative purposes**:
  – C is the minimum grade in this class if you want to take the JAVA programming class.
  – A or A- average is recommended if you want to be a CS major

• **Take this class Pass/Fail (deadline March 30) if**:
  – Your worries about the grade are getting in the way of the learning process and you are not planning to be a CS major or minor
Your Middle-of-the-Term Grade

• Purpose of Grade: Advisory Only
  – Will not go on transcript

• Grade will be calculated as follows:
  – def middle_grade(mid1, quiz, hw):
    ## mid1 = Midterm1 grade 0-4
    ## quiz = Average Quiz 1-7 converted to 0-4
    ## hw = Average HW 1-5 converted to 0-4
    grade = (.05 * quiz) + (.4 * hw) + (.55 * mid1)
    return(grade)

• Different than final grade:
  – HW has only .2 weight on final grade
  – There are 3 tests instead of 1
  – There will be an extra credit opportunity
Printing and Strings

• A string is an object consisting of characters and represented in Python as being surrounded by quotes, e.g.,
  – 'TThherre iss nnotthhinngg tto ffearr!!'
  – ' ---\n|Fox|\n ---'
  – ' \\\ \ /'

• Printing a string causes the representations of the characters to be interpreted, including escape characters like '\n' and '\\'
  – TThherre iss nnotthhinngg tto ffearr!!
  – ---
    |Fox|
    ---
  – \\ \ /
“Break” vs “Pass”

- **break** – causes a loop to exit
  - num = 1
    while True:
      print(num)
      num = num + 1
      if num>5:
        break

- **pass** – do nothing under if/elif/else
  - def identify_duck(animal):
    if animal == 'dog':
      pass
    elif animal == 'goose':
      pass
    elif animal == 'duck':
      print('Yes')
Correct Use of input function

• `input(string)` returns whatever user enters

• Example:
  – `answer = input('Why did the chicken cross the road?')`
    • First 'Why did the chicken cross the road?' prints
    • Then user enters whatever they want
    • Then the variable `answer` is set to whatever the user typed.

• On tests, the connection between the variable `(answer)` and the input statement was not always clear.
Tests and Answers

• Test Version1 (Section 009)
  – Test Part1: https://cs.nyu.edu/courses/spring18/CSCI-UA.0002-004/midterm1_version1_with_corrections.pdf
  – Answers: https://cs.nyu.edu/courses/spring18/CSCI-UA.0002-004/midterm1_version1.py

• Test Version2 (Section 004)
  – Test Part2: https://cs.nyu.edu/courses/spring18/CSCI-UA.0002-004/midterm1_part2_version2.pdf
  – Answers: https://cs.nyu.edu/courses/spring18/CSCI-UA.0002-004/midterm1_version2.py
Middle of Term Grades

• 0.05 * Quizzes + 0.45 * HW + 0.5 * Midterm
• Unofficial Grade – will not go on your transcript
• Purpose: Advisory Only