Introduction to:
Computers & Programming:
Administrative Matters

Adam Meyers
New York University
Outline

• What to Expect from a Intro to CS Class
• Basic Info:
  – Class Schedule, Room Number, Office Hours, Contact Info, Course Webpage
• Policies:
  – Homework, Exams, Grading, Cheating
• What is covered in the Class
  – Materials, Syllabus, Structure of classes
• Installing Python
What to Expect from Intro to CS

• You will learn how to write computer programs:
  – Sets of instructions that computers follow to do solve problems, display images, etc.
  – The end result of creative problem solving

• Skills that are helpful:
  – Ability to follow instructions loosely
  – Experimentation and game play when solving problems
  – Combination of mathematics (especially logic) and creativity
  – Asking questions without fear
  – Willingness to fix things when they don't work (and not being afraid of breaking something).

• My best students varied a lot in their specialties: mathematicians, artists, filmmakers, philosophers, musicians, etc. …
What to Watch Out For

- Computer Science is a type of Math: info is sequential
- Key concepts can hold you back if you don't understand
  - Similar to disciplines including: art, music, math, …
- May be more difficult than a typical “Intro to” Elective
- Do not start this class late in the term
- **Do not hesitate to ask questions in class**
- **Do not do homework late or miss classes (lectures or labs)**
  - **This can have a substantial effect on test grades**
  - There is no (official) penalty for missed classes or late HW (until end of the term), but doing so impedes understanding
- If you don't understand, seek help immediately so you don't fall behind
  - Ask tutors (lab classes, computer labs, etutors)
  - Ask me (office hours, appointment, email)
- If you have trouble & are grade conscious, consider taking class Pass/Fail
Basic Info: CSCI-UA.0002 Fall 2015

• Websites
  – Section 004: http://cs.nyu.edu/courses/spring16/CSCI-UA.0002-004/
  – Section 009: http://cs.nyu.edu/courses/spring16/CSCI-UA.0002-009/
  – CSCI-UA.0002 Website:
    • http://cs.nyu.edu/courses/spring16/CSCI-UA.0002-003/common_syllabus/

• Schedule:

<table>
<thead>
<tr>
<th>Section</th>
<th>Days</th>
<th>Time</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>004</td>
<td>M &amp; W</td>
<td>8:00-9:15AM</td>
<td>CIWW 102</td>
</tr>
<tr>
<td>009</td>
<td>T &amp; TH</td>
<td>8:00-9:15AM</td>
<td>CIWW 312</td>
</tr>
</tbody>
</table>

  – No Classes: Monday February 15 and Spring Break (March 14-20)
  – Midterm1: Feb 29 (Section 004) and Feb 25 (Section 009)
  – Midterm2: Apr 11 (Section 004) and Apr 7 (Section 009)
  – Final: May ?? (Section 004) and May 17 (Section 009)

• My office: 715 Broadway, Rm 702
• Office Hours: Monday: 1:30-3PM or Thursday: 10:30-12PM or by appointment
• My Email: meyers@cs.nyu.edu
• My Phone Number: 212-998-3482
Website and NYUClasses

• NYUClasses
  – Submitting Homework
    • Usually python code or text
    • Non-programming HomeWork Should be in pdf form (e.g., scan of drawn flowchart)
  – Getting Grades
  – Course Documents with Licensing Restrictions

• Website – All Other Course Information
  – Schedule, class lectures, sample test questions, python information, etc.
  – Quizzes and supplemental videos
  – Updated throughout the semester (errors fixed, new stuff added)
Grades

- Homework 25% = 5% quizzes + 20% prog. assignments
- Midterm 1 20%
- Midterm 2 20%
- Final 35%
- All grades are converted to letter grades before averaging (A = 4, A- = 3.7, B+ 3.3, ..., D = 1, F = 0)
Your Grade in Python

def calc_grade(Quizzes, Prog_Assts, Mid1, Mid2, Final):
    # Quizzes is the average grade for quizzes
    # Prog_Assts 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
    HW = (.05 * Quizzes) + (.20 * Prog_Assts)
    Grade = (HW * .25) + (Midterm1 * .20) + (Midterm2 * .20) + (Final * .35)
    return(Grade)
Cheating

• Tests, homework and grades are:
  – Tools for assessment of what you did and did not learn
• Cheating is when you copy someone else's work without giving them credit.
• Discussing homework with classmates is not cheating if:
  – Each student produces a different solution and/or
  – Afterwards, each student could solve similar problems on their own
• Consulting other students on tests is definitely cheating.
• The only reason to cheat is:
  – You think obtaining high grades is more important than learning
    • And you don't think you will be caught
• I have no respect for cheaters:
  – They will fail the class and be reported to the administration
Late Homework, Missing Classes and Grades

- It is Easier to Learn the Material in Class (including labs)
  - Reading online lecture notes is not the same as attending
  - I sometimes present material differently from the modules and from the book
    - For example, I place greater emphasis on writing code using functions
    - This can make relying on online materials more difficult
- Multi-choice quizzes – no credit if late
- Other Homeworks: Lateness → Natural Consequences, Fairness and Guilt
  - Natural Consequences
    - Easier to prepare for tests if you have graded homework on the material on the test
    - Doing homework on time makes it easier to keep up because CS is sequential
  - Fairness: The graders mark on-time homework first and then late homeworks
  - Guilt:
    - Graders have lives too – it is an imposition on them when you do late HW
    - They are paid by the hour – if they run out of time, they may not have time to grade your late homework, e.g., they have to pass their classes too
    - Extremely Late HW may never be graded if the grader simply does not have time
  - No Points Taken Off for these unless it is the End of the Term and Grader simply does not have time
  - I think these consequences are severe enough without me taking additional points off, which I think would compound the problem
If You Miss a Class, Please keep Up With the Material

• Check the website
• Pay me an office visit
• Consult with a classmate (get their contact info)
• Plan ahead so you can go to the corresponding class in another section (ask me first, please.)
Homework

• Represents 25% of your grade
• Chapters in the Gaddis Book
• Online multiple choice quizzes after each chapter (5%)
• Programming Assignments (20%)
  – Each program is judged as follows:
    • Does it solve the problem posed by the assignment
    • Is it clear?
    • Does it work?
    • Do you go beyond the assignment and show that you really know what you are doing? Do you add “bells and whistles”?

• Supplemental Readings
  – From the Internet
Why Does it Matter That I am Teaching 2 Sections

• It is possible to makeup missed classes and exams
  – It is OK to switch classes occasionally, to make up a test or a missed class, etc. as an exception to the rule
    • provided that you let me know (by email) and one class does not become overcrowded and disfunctional.
  – In terms of make-up classes, the alignment may be imperfect, e.g., if one class ends up going more quickly than the other for any reason (e.g., fire drills, bad weather, etc.)
• Shared office hours (with a 3rd class on another topic as well).
Syllabus

• Introduction to Programming Languages
• Python Basics
• Working with variables and operators
• Control Structures
• Repetition Structures
• Working with Text
• Functions and Modules
• Lists and Dictionaries
• File Input and Output
• Other topics, as time permits
Structure of Classes

• **Types of Classes**
  – 1st class most weeks: lecture class, sometimes with in-class program
  – 2nd class most weeks: lab-style class.
    • Everyone will do their “homework” in class with the support of T.A.s and myself
    • Finish Homework at Home and send in (approximately 1 assignment per week)
  – Tests: midterm1, midterm2 and final
  – Reviews for Test

• **Laptop Recommended, but if you do not have a laptop**
  – We will partner you with someone who does and you will both include each other's names on your homework so the grader knows who you partnered with.
  – You will be able to get copies of your work by email and submit it later.

• **Online materials (weekly):**
  – Online short videos on specific Topics (see website)
  – Online quizzes paired with reading material
Materials

- *Starting with Python (Third Edition)* by Tony Gaddis
  - Cover most (not all) material in book, as per HW assignments
  - Rate of reading: 1 chapter every 2 or 3 classes
- **Self-Study Materials on Website:**
  - Online short videos on specific Topics
  - Online quizzes paired with reading material
- **Slides and Notes from Lectures:** most material available online, but copyrighted material in NYUClasses (just 1 set of slides)
- **Python Documentation** (reference): [https://docs.python.org/3/](https://docs.python.org/3/)
- **Additional material:** How computers work, algorithms, properties of other programming languages, etc.
Install Python (Part of 1\textsuperscript{st} Homework)

- Go to \url{www.python.org}
- Click on the Download tab
- Download Latest Version of \textbf{Python 3} (NOT Python 2)
  - Version 3.4.3 as of June 26, 2015
  - For Windows, MAC or Linux
    - * Python 3 for tablets & cell phones exists, but is not handled in this class and not currently supported by NYU tutors or professors
- Install Python
  - If there are installation problems, do not be silent
  - Additional Instructions at: \url{http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-002/common_syllabus/#software}
  - If you have a laptop, take your laptop to the \textbf{lab tutors}
    - Weekdays in the glass rooms in back of labs at 14 Washington Place
  - Other tutoring available – see common curriculum website:
    - \url{http://cs.nyu.edu/courses/fall15/CSCI-UA.0002-002/common_syllabus/}