CS 111: Program Design I Lecture 2: Intro cont. & Python Basics

Robert H. Sloan & Richard Warner
University of Illinois at Chicago

August 29, 2019

UIC

ACTIVE LEARNING & CLICKERS

Lecture: Partially Peer Instruction

- Pose carefully designed question
  - Solo vote: Think for yourself and select answer
  - Discuss: Analyze problem in small teams
    - Practice analyzing, talking about challenging concepts
    - Reach consensus
    - If you have questions, raise your hand and I will come over
  - Class wide discussion:
    - Led by YOU (students) – tell us what you talked about in discussion that everyone should know!

Why Peer Instruction?

- You get to make sure you are following the lecture.
- We get feedback as to what you understand.
- It’s less boring!
- Research shows it promotes more learning than standard lecture. (Mazur, Harvard)

Discussion Groups

- 2–4, ideally 3 people each
- Ad hoc for at least first few classes until enrollment settles
- May assign groups later on

Discussion Etiquette

- It’s okay to not know things
  - If you already know everything, you’re wasting your time
- It’s not a competition

Example: The best dinosaur is:

A. T-rex

B. Raptor

C. Triceratops

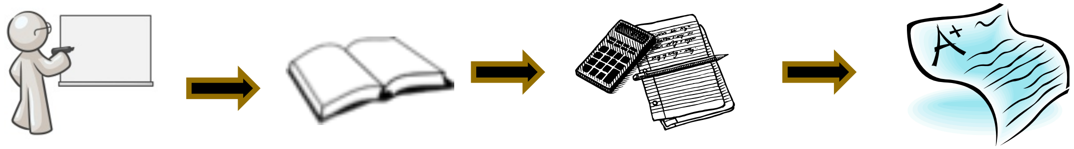
D. Stegosaurus

E. Some other dinosaur

Doing The Reading Means

- You do the “easy” part before class.
  - Read it and analyze for YOURSELF!
  - If I rephrase it for you, what purpose does that serve?
- Traditional class structures often look like:

Lecture Textbook Assignment Exam



First Exposure

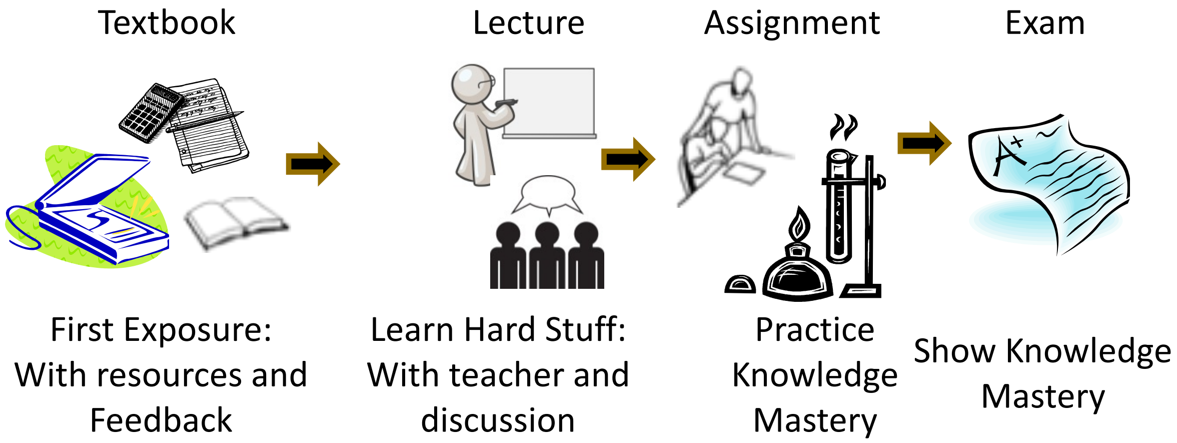
Read Hard Stuff

See if You Know Hard Stuff

Show Knowledge Mastery

- You get very little opportunity for “expert” feedback

Active Learning



Textbook

First Exposure:
With resources and
Feedback

Lecture

Learn Hard Stuff:
With teacher and
discussion

Assignment

Practice
Knowledge
Mastery

Exam

Show Knowledge
Mastery

- Greater opportunity for expert feedback!
- Research on how people learn:
  - Everyone constructs their own understanding
    - I can’t dump understanding into your brain
  - To learn, YOU must actively work with a problem and construct your own understanding of it

Does this mean I have to show up to class?

- **Yes**

- Will drop your lowest 3 class participation scores

Collaboration Policy

- Please do
  - Post on Piazza forum
  - On Piazza, post relevant small snippet of your code to discuss code problem, not your whole program
  - Talk with classmates about assignments
- Please don't
  - Copy someone's code
  - Show someone else your code
- Good rule of thumb: Wait half an hour after discussing, write code/do problem

A BIT MORE RE SYLLABUS ETC.

Welcome to UIC CS

- CS Student Affairs Office (905 SEO)
  - Visit with any questions. If they can't answer, they can tell you who can.

- Director of Undergraduate Studies
  - Joe Hummel (908 SEO)

Syllabus in a Nutshell

- Office Hours on the website
- Piazza Q&A system

- Labs, Wednesdays in SEL

- Programming work for the course
    - Lab Problem
        - A few will finish in Lab; some need more time, due Thu., 11:30 pm
        - Almost every week
        - Lowest two dropped; *cannot drop first one*
            - *Want to be sure you've done one before 10th day drop deadline*
    - Homework/Programming *Projects*, specified deadline, usually later
        - About 4 to 8 total

    - Three late day passes, *valid only on projects*

Reading still open

- 1st Zybooks reading assignment and lab deadline extended until Monday night, 11:30 pm to accommodate students switching sections
- 2nd reading assignment that will be due Tuesday by 1:30 pm likely to be posted soon (or is already posted)

Grading: See syllabus for fine details

Lab programming assignments (lowest 2 dropped)	18%
Lab quizzes (lowest 2 dropped)	5%
Programming Projects	22%
Two midterms, 10% each	20%
Final exam	20%
Lecture participation (clicker), Zybooks participation activities, Zybooks challenge activities, 5% each	15%

In addition, to pass CS 111, *one must pass both the programming part of the course (labs plus programming projects) and the exam part of the course (midterms plus final).*

No laptops, smartphones, or tablets

- Worse for you
- *Negative externality: Your* use results in *other students* having worse results
- *See* "The Pen Is Mightier Than the Keyboard: Advantages of Longhand Over Laptop Note Taking", *Psychological Science*, 2014; *New York Times,* November 22, 2017

Questions?

- All of this info and grading policy, etc., etc., on course website
- (Show the beautiful course website)

How to Ace This Class

- Do the reading
- Attend class and participate in discussion
- Start programming projects early
- Go to lab section and ask questions
- If you get stuck, post on piazza or come to office hours
- Come to office hours also if you’re not stuck!
- Have fun!

How do I become a great computer scientist?

- Write a lot of code!

PROGRAMS, PROCESSES, ALGORITHMS

Which is correct?

A. A computer program is typically a more detailed fleshing out of the general specification of an algorithm

B. An algorithm is typically a more detailed fleshing out of the general specification of a computer program

C. Computer programs & algorithms are typically stated with the same level of detail

Programming is a communications and analysis skill

- If you want to understand what your tools (e.g., Excel) can or cannot do, you need to understand what the programs are doing
- If you want to say something that your tools don’t allow, program it yourself
- If you care about analyzing data… then it’s worth your while to understand how to do it at scale!
- Knowledge is Power, Knowing how programs work is powerful and freeing

Aside to non-CS-Majors: *Process*

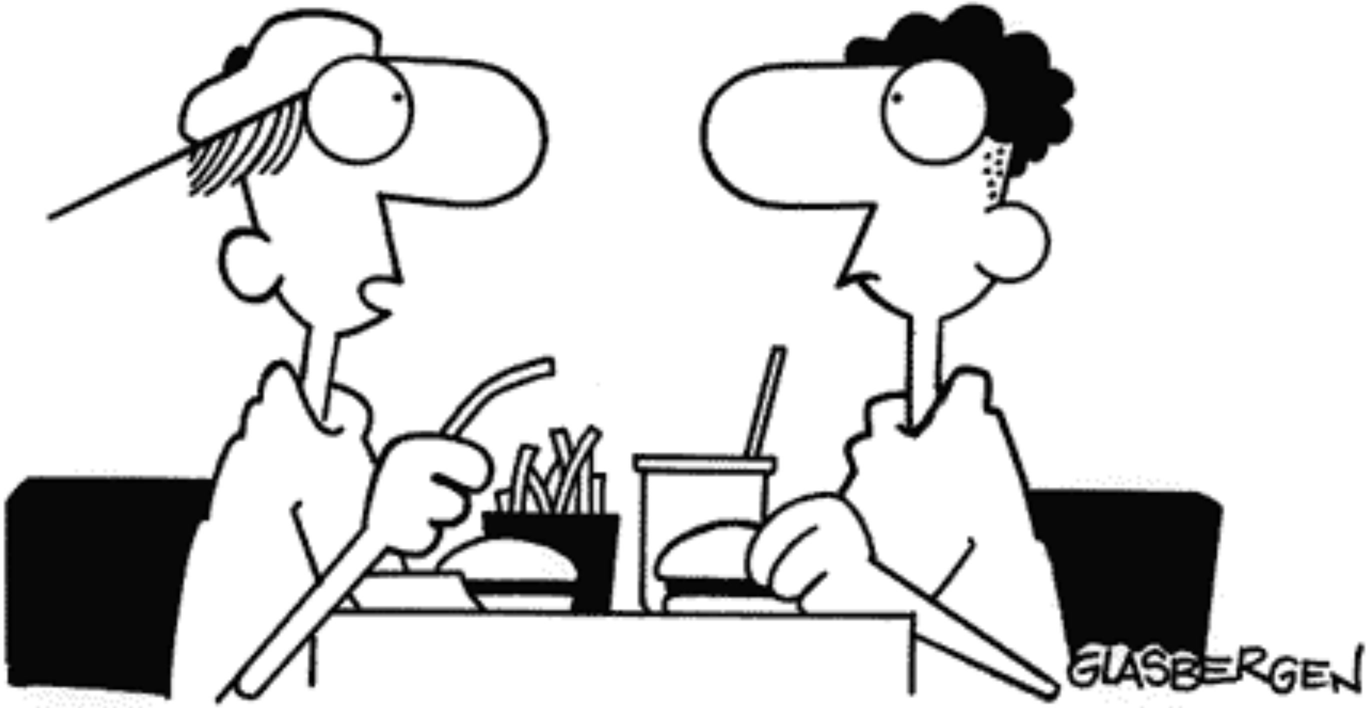
- Alan Perlis
  - One of the founders of computer science
  - Argued in **1961** that Computer Science should be part of a liberal education: *Everyone* should learn to program.
    - Perhaps computing is *more* critical to a liberal education than Calculus
    - Calculus is about rates, and that’s important to many.
    - Computer science is about **process**, and that’s important to *everyone*.

Finally: Programming is about Communicating Process

- A program is the most concise statement possible to communicate a process
  - That’s why it’s important those who want to specify *how* to do something understandably in as few words as possible
  - And one reason why we should strive to make our programs easy for *humans* to read

So let’s start!

©1997 by Randy Glasbergen. E-mail: randyg@norwich.net
http://www.norwich.net/~randyg/toon.html



“I forgot to make a back-up copy of my brain, so everything I learned last semester was lost.”

LET’S MEET PYTHON

Let’s meet Python

- And use it for our first ~4 week mission:
- Learn to write encryption functions used to enable secure, secret electronic communication
  - So we can carry out criminal and terrorist activities *without*
  - the Vague, Yet Menacing, Government Agency (VYMGA) knowing what we said

Let’s meet Python

- And use it for our first ~4 week mission:
- Learn to write encryption functions used to enable secure, secret electronic communication
  - So we can carry out ~~criminal and terrorist activities~~ secure conversations with our banks, our lovers, and our start-up collaborators *without*
  - the Vague, Yet Menacing, Government Agency (VYMGA) knowing what we said

Let's meet Python

- And use it for our first ~4 week mission:
- Learn to write encryption functions used to enable secure, secret electronic communication
  - So we can carry out ~~criminal and terrorist activities~~ secure conversations with our banks, our lovers, and our start-up collaborators *without*
  - ***anyone*** ~~the Vague, Yet Menacing, Government Agency (VYMGA)~~ knowing what we said

A few words about why Python

- Popular, widely used (there are jobs!)
  - Top 5 for overall general use
  - #1 or #2 for data science/data analytics (with R)
- Easy (for humans!) to read; easy to write; easy to learn
- Can do realistic examples very early on
- Outstanding for exploratory, experimental, "get an answer" programming
- Twice the fun for half the annoying syntax!

Encryption, Privacy, and Government Power

A Limit on the Government’s Power

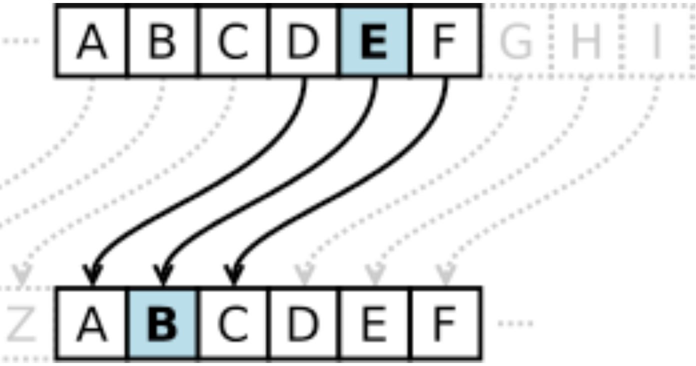
- "The poorest man may in his cottage bid defiance to all the force of the crown. It may be frail—its roof may shake—the wind may blow through it—the storm may enter, the rain may enter—but the King of England cannot enter—all his force dares not cross the threshold of the ruined tenement."
  - William Pitt, speaking in the British Parliament in 1763
- Why?
- To make sure privacy limits the power of the government.

A Modern Power Question

- How do you ensure privacy
  - When we store vast amounts of personal information online?
  - When virtually all digital transactions are recorded?
- You can try encryption.
  - Susan Landau, *Listening In: Cybersecurity in an Insecure Age*

Example: Caesar's Cipher

- Key = 3 (right shift of 3)
  - (According to Suetonious, *Life of Caesar*)
- E("HELLOBOB") = KHOORERE
  - Note: Reportedly fairly secure in Julius Caesar's time, when mere literacy was rare
- Decrypt:



The FBI Wants Your Key

- Suppose you encrypt files on your laptop with a Caesar cypher with a key of 3.
- Then the FBI serves you with a subpoena demanding that you give them the files and the encryption key.
  - A subpoena is a court order.
    - ‘Subpoena’ is Latin for “under penalty”. The FBI would serve you with a subpoena duces tecum (= “under penalty to bring with you”): an order to produce the key.

Easy To Get A Subpoena

- You can get documents necessary to prove issues in a case
- as long as the information is not “privileged”
- and not available by any other means.

Suspicion of Wrongdoing

- In *U. S. v. Doe*, Doe used a YouTube account the FBI suspected of exchanging child porn.
- The FBI determined that he accessed the Internet hotel rooms, so it tracked him to a hotel room, arrested him.
- It searched the room finding two laptops and five external hard drives.
- They could not decrypt the drives and so they subpoenaed the encryption key.
- Should Doe have to give up the key?