Computer Science I: Fundamentals (CS:1210)

SPRING 2014

INSTRUCTOR: SRIRAM PEMMARAJU

TEACHING ASSISTANTS: SIKDER HUQ, NIRAJ PANDYA, NAVEEN DEWDA, RICHA GANDHI

CONGRATULATIONS!

- This is an exciting time to be a computer scientist!

- **Computational Thinking** is becoming part of all aspects of life:

  - **Biology, Healthcare, Pharmacy, Biomedical Engg**
  - **Physics, Astronomy**
  - **Economics, Psychology**
  - **Music, Film**
  - **Humanities**
  - **…**

Core CS Areas are Thriving!

- Data Mining
- Graphics
- Human Computer Interaction
- Networks and Distributed Computing
- Natural Language Processing
- Vision
- Robotics
- Algorithms

Here is ACM's poster on careers in computing:

http://www.acm.org/membership/careernews/extras/careercolor.pdf

Search Engines

- If you've searched the web lately using *Google*, *Bing*, *Baidu*, etc., you've used one of the most impressive contributions of Computer Science.

- How do search engines manage to search through *billions* of web pages spread across the world and find just those you were looking for in microseconds?

- Advances in *Algorithms*, *Data Mining*, *Distributed Computing*, *Networking*, etc., all come together in search engines.

Watson versus The Humans

- In Feb 2010 "Watson" the Jeopardy playing machine built by IBM handily defeated human champions Ken Jennings and Brad Rutter.

- Read a pre-match article at Wired:
  http://www.wired.com/epicenter/2011/01/ibm-watson-jeopardy/

- Watch an edited video of the show at:
  http://www.youtube.com/watch?v=YLR1byLoU8M

- This is the result of major advances in *natural language processing*.

Microsoft Xbox Kinect

- A controller-free gaming add-on to Xbox 360

- The Kinect sensor does full-body 3D motion capture, facial recognition and voice recognition.

- The software does motion analysis with feature extraction of 20 joints per player.

Computational Epidemiology at Iowa

- Computational tools to model, simulate, visualize and understand the spread of disease.

- Goal is to provide information to general public, hospital policy makers, etc.

- We use *Algorithms and Graph theory, Data Mining, Sensor Networks, Statistics, Visualization,...*

- Visit http://compepi.cs.uiowa.edu/

Other amazing examples...

- Google's project on "self-driving" cars. See this article in Spectrum: http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works

- The success of Mars exploration rovers. See this page at Jet Propulsion Labs (JPL):

  http://marsrover.nasa.gov/home/index.html

Computer Science I: Fundamentals

- Is much more than programming…

  (We will repeat this many times.)

- A successful student will learn to view the world through a "computational lens."

- Introduction to
  - Techniques for solving computational problems
  - Designing algorithms
  - Thinking about their efficiency
  - Translating algorithms into reliable, reusable software

Be warned

- Most students will find this course challenging!

   (Not because this is a "weed out" course or because the instructor is ill-tempered.)

- Designing and implementing algorithmic solutions to computational problems requires a new and possibly unfamiliar way of thinking.

- Peter Norvig, Director of Research at Google, talks of "years of deliberative practice" to become of proficient programmer in his article "Teach Yourself Programming in 10 Years."

- As instructors, our role is to design a program of "deliberative practice" for you and to push you to work through this program independently.

To be successful…

- This should be the only course you are taking this semester! ☺

- Separate **algorithm design** and **coding**.

- Stay **unplugged** as much as possible.

- Program **incrementally**. **Test** each increment before moving on.

To be successful…

- Attend lectures and attend your discussion section (on Tuesday) regularly.

- Help is plentiful; so see us often. Information on office hours and coordinates will be available on the course.

- Visit the course website
    http://www.cs.uiowa.edu/~sriram/16/spring14/
  regularly (e.g., at least once a day). All lecture notes, homeworks, quizzes, solutions, announcements, etc., will appear there.

- Turn in all your work on time. *Late submissions will not be accepted except for officially sanctioned reasons.* (We will use ICON dropbox for submissions.)

Overview of syllabus

- See the syllabus document for all kinds of information including
  - Topics we will cover and a schedule
  - The components of evaluation
    - quizzes (weekly in discussion sections and a few pop quizzes in lectures)
    - homeworks (six, biweekly)
    - Programming projects (two, 3-4 weeks long each)
    - Exams (three, two in-class and one final)

  - Policies on cheating, classroom ettiquete, communicating with instructors and TAs, etc.

Students with disabilities

I would like to hear from anyone who has a disability which may require seating modifications or testing accommodations or accommodations of other class requirements, so that appropriate arrangements may be made.

Please see an instructor right away. Also visit Student Disability Services website for more information.

Onto an unpleasant matter…

- Quizzes, Homeworks, Projects, and Exams are meant to test each *individual* student's mastery of the material.

- You cannot pass off someone else's work as your own. This is cheating.

- Giving a fellow student solutions is also cheating.

- You can talk and discuss ideas and concepts, but no actual exchange of written material, code snippets, answers, etc.

- If you are not sure, see the instructor right away.

Programming Language: Python

| Pascal | → | C++ | → | Java | → | Python |
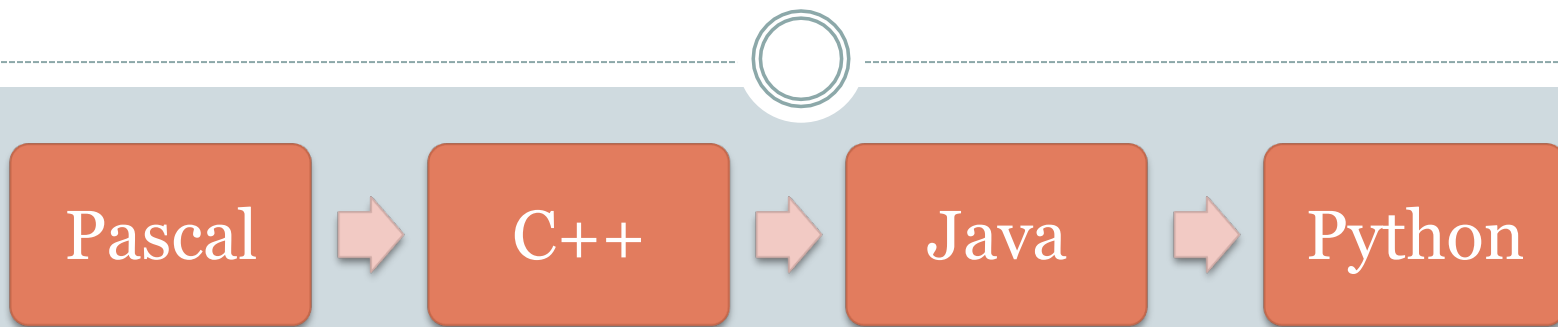
Language in CS I over the years

A pitch for Python

- Easy to get started
- Allows beginners to focus on getting the computer to do what they want!
- Interactive mode is great for experimenting
- Extensive standard and third-party libraries
- No variable declarations, run-time rather than compile-time errors

To do right away...

- Become familiar with a Python programming environment (IDE).

- Two popular IDEs for Python are:
  - IDLE: open source, usually bundled with Python
  - Wing: proprietary, developed by Wingware, free version available for download for Windows/Mac OS/Linux.

- Examples in this class will use the Wing IDE.

- Instructions for downloading Python and Wing IDE for Windows and Mac OS are also available on the course page.

Have a good semester!

- See you on Friday.