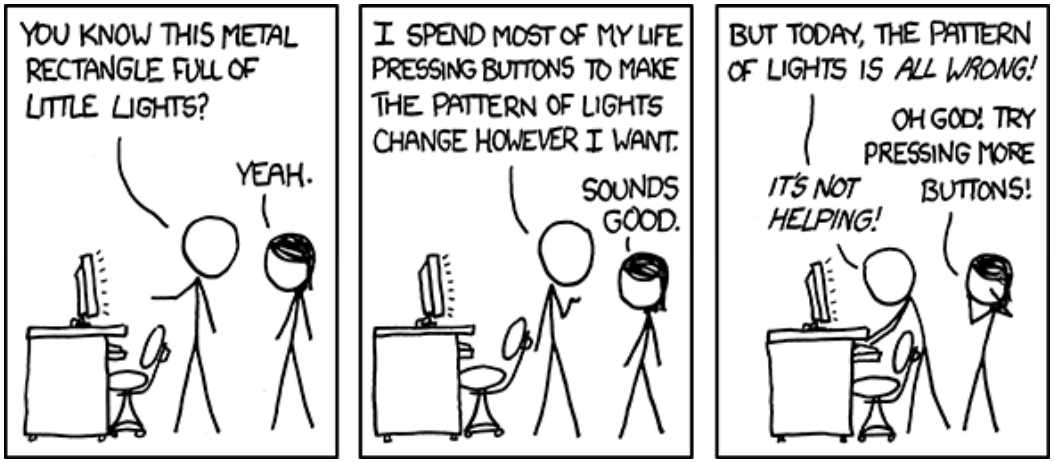
CS 161
Introduction to CS I

- What this class offers you
- What your responsibilities are
- How to communicate with computers



Randall Munroe, xkcd.com

Oregon State University
College of Engineering

Ph.D. in Computer Science

Jet Propulsion Laboratory
California Institute of Technology

M.S. in Geology

Oregon State University
College of Engineering

MLIS in Library and Information Science

Airplane pilot

Oregon State University
College of Engineering

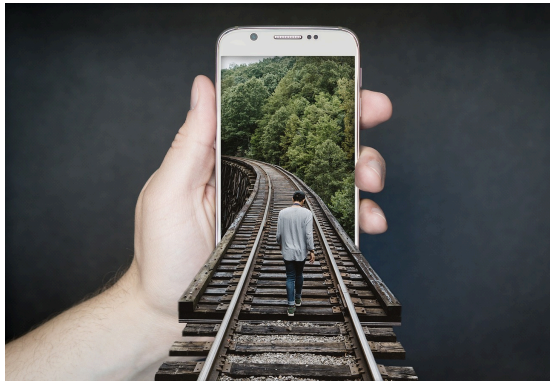
About you

- First year at OSU?
- First quarter at OSU?
- Prior experience with programming?
- Prior experience with C++?

Oregon State University
College of Engineering

Start with why

- Why are you here?
  - Build apps, games, simulations, robotics, biology, AI, ML, …
  - What can computer scientists do to improve the world?



Oregon State University
College of Engineering

How do you become a computer scientist?

- Technical skills: use of tools
  - Programming
  - Testing
  - Debugging
- Conceptual skills: logic and creativity
  - Problem solving
  - Algorithm design
  - Analysis – efficiency, ease of use, what is computable?

Course map

Divide and conquer part 2
(recursion)

Structured data
(arrays)

Basics
Storing data, calculations, interacting with users

Decision making (adaptation) and **repetition** (write once, repeat forever!)

Divide and conquer
(modularization and code re-use in functions)

Dynamic growth
(memory allocation and management)

Oregon State University
College of Engineering

Visit the course website

http://classes.engr.oregonstate.edu/eecs/winter2020/cs161-020/

CS 161 - Introduction to Computer Science I

Winter 2020: MWF 2 - 2:50 p.m., LINC 228

Home | Syllabus | Calendar | Assignments | Labs | Useful Links | Student Clubs | TA Bios

Important Dates:

01/06 - First day of class
01/12 - Last day to Add a class without dept. approval
01/12 - Last day to Drop a class for 100% refund
01/20 - Martin Luther King, Jr. Day (No School)
03/16 - Final Exam, 6-7:50 p.m. in LINC 228

Instructor: Dr. Kiri Wagstaff
2079 Kelley Engineering Center
kiri.wagstaff@oregonstate.edu
(541) 737-9676
Office Hours (2079 KEC): **Mon 4 - 5 p.m. and Weds 3 - 4 p.m.**

Graduate TAs: Sabrina Jesmin and Yipeng (Roger) Song
jesmins@oregonstate.edu and songyip@oregonstate.edu
Office Hours: See Below

Oregon State University
College of Engineering

Syllabus highlights (1)

- **You are responsible for following all course policies and info in the syllabus.**
- Attend lecture. Missed in-class work cannot be made up.
  - Check the calendar for **assigned readings** and **assignment due dates**
  - Silence cell phones in class
- Attend lab: your chance to get hands-on practice!
  - Lab activities cannot be made up without prior approval
  - Jan. 20 – attend another lab or complete outside lab (checked off on Jan. 27)
- Assignments: Sign up to demo your work **within 2 weeks after due date** (earlier is to your advantage)
  - **Submitted code must compile** (else 0 grade for coding part).
    Comment out or remove any parts that prevent compilation (for partial credit).
  - **Take notes during demo.** Write up ideas for improvements = extra credit.

Oregon State University
College of Engineering

Syllabus highlights (2)

- Course grade:
  - 40% - 5 assignments
  - 10% - 10 labs
  - 10% - designs + peer reviews
  - 30% - 2 midterm exams
  - 10% - final exam (cumulative)
- Proficiency demo (week 10)
  - Must pass to maintain a passing grade in the class
  - Practice demo in week 5 so you know what to expect

Oregon State University
College of Engineering

Syllabus highlights (3)

- Getting help: Re-read assignment, textbook, Piazza, TAs, instructor, tutors
  - See guidelines on Email Etiquette
  - My office hours: Mon 4-5 p.m. and W 3-4 p.m., KEC 2079
- Course buddies
  - Strength in numbers!
  - Growth mindset: we are all learning and can help each other
  - Understand when to collaborate/consult and when to work solo

Oregon State University
College of Engineering

Guest speaker: Casey Patterson

- OSU COE Student Success Coordinator
  - Use free tutoring hours: https://engineering.oregonstate.edu/current-students/academic-support/undergraduate-tutoring
    - Link available on our course website on the "Useful Links" tab, under "Need help?"
  - OSU has a process for handling academic misconduct
    - Be familiar with student code of conduct to know what is allowed
    - In this class: you can discuss problems, assignments, ideas, but **all code and written answers you submit should be your own**

Houses

Charles Babbage

Ada Lovelace

Grace Hopper

Alan Turing

Course map

Divide and conquer part 2
(recursion)

Structured data
(arrays)

Dynamic growth
(memory allocation and management)

Basics
Storing data, calculations, interacting with users

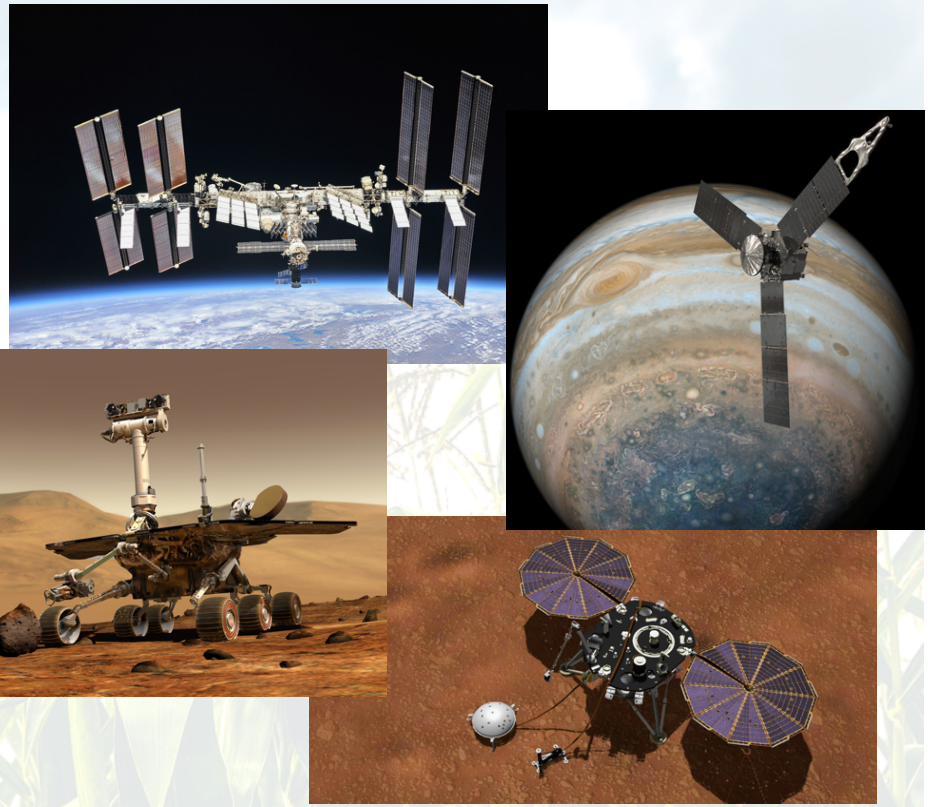
Decision making (adaptation) and **repetition** (write once, repeat forever!)

Divide and conquer
(modularization and code re-use in functions)

Oregon State University
College of Engineering

Computers in our lives

- How many are in this room?
- They're also in space and on other planets

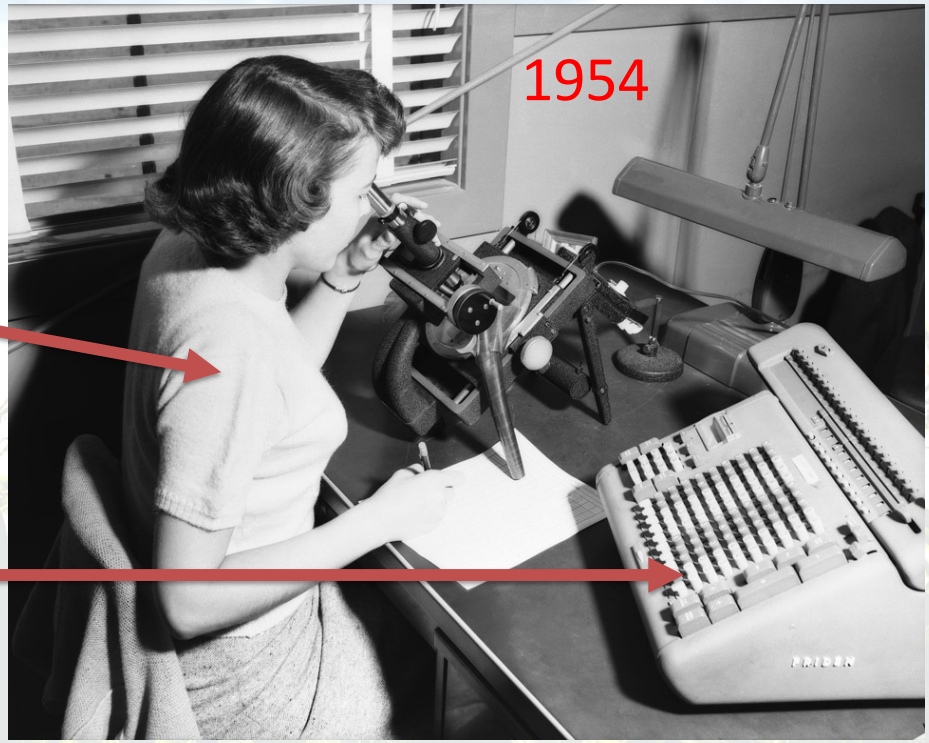


Computers in our lives

- What is a computer?

Computer
(a person's job)

Calculator



Computers in our lives

- What is a computer?
  - "Computer: an electronic device for storing and processing data, typically in binary form, according to instructions given to it in a variable program." (Oxford)

Communicating with computers

- Not there yet: https://youtu.be/LkqiDu1BQXY?t=64

C++ (code) → Compiler (translator) → Computer

– Think about this class as a foreign language class

– What strategies work for you when learning a foreign language?

What ideas and skills did we learn today?

- Approach learning to program like learning a foreign language
  - Practice is your most powerful tactic!
- Programming process
- Tools: editor, compiler (later: debugger)
- Submit your own work. Ask for help from course staff anytime!
  - Piazza discussions, TAs, instructor, tutors

Oregon State University
College of Engineering

How to succeed in this class

- Start with why
- Get lots of practice writing your own programs
  - You are your own best teacher. Experience makes the best programmers.
- If your lab is early in the week, read ahead in **Rao** to prepare for lab.
- If your lab is late in the week, read the **lab** in advance and practice on your own so you are prepared for assignments.
- Use office hours. If something isn't clear, ask questions.
- Use tutoring hours (**get extra credit!**)
- Be proactive (e.g., accommodate absences, other issues early)
- Take good care of yourself: sleep, food, exercise, breaks

You are ready for week 1!

- ❑ Attend and complete **lab** (laptop required)
- ❑ Read **Rao Lesson 1** (pp. 1-15) **and Lesson 2** (pp. 17-29)
- ❑ Try **Rao Exercise 2.1** (p. 29) – answers at the back of the book
- ❑ Get started on **Assignment 1** (due **Sunday, Jan. 12**)
  - Don't wait until your lab to start working on it. Reading, designing, thinking, and planning do not require access to an editor or compiler.

See you Wednesday for more adventures!

- ❑ Bring: the **number of light switches** in your home
- ❑ Bring: scratch paper and writing utensil