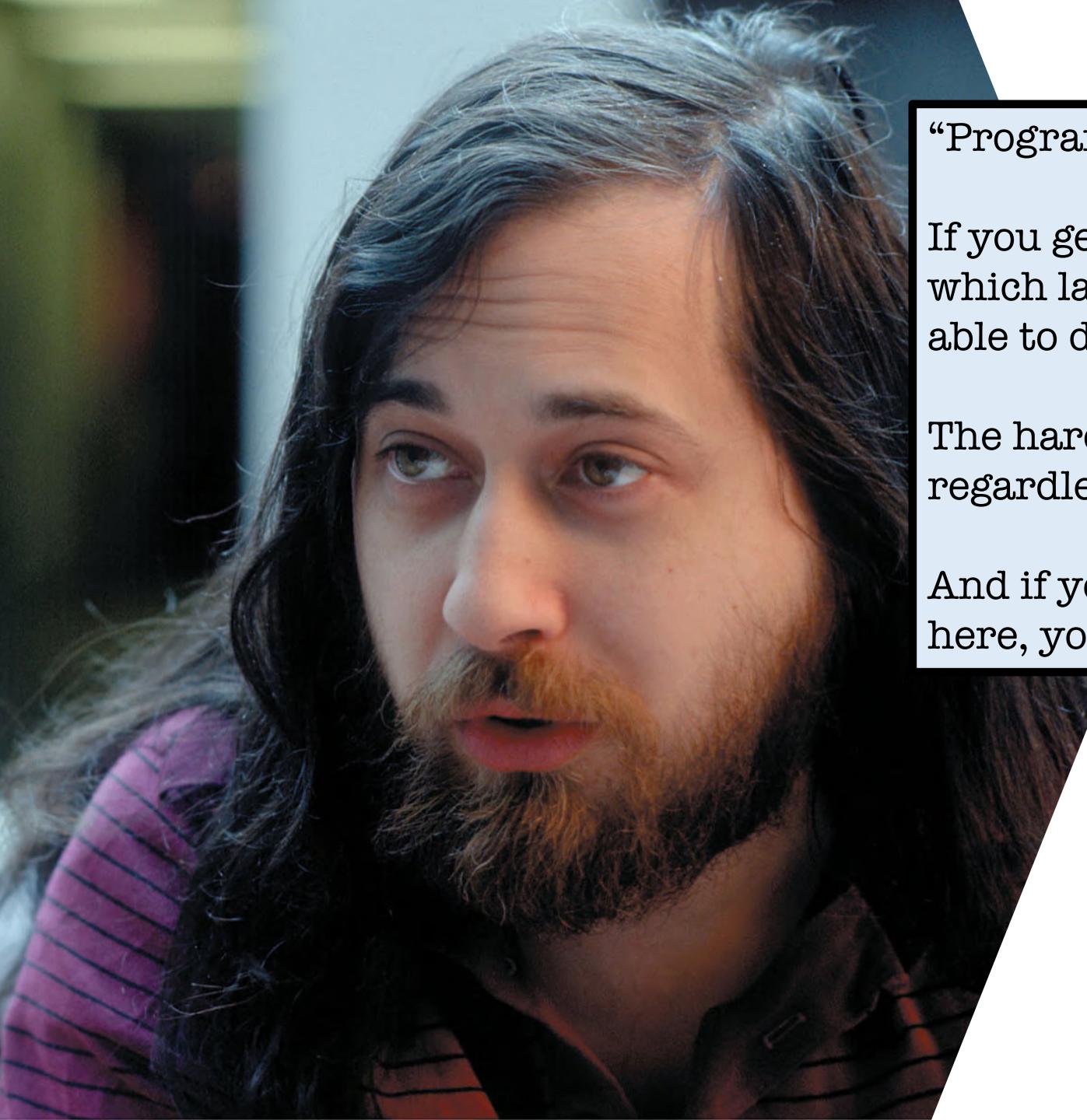


**"We were sure that this machine could do anything we wanted it to do.
We were very cocky about that."**

Marlyn Wescoff Meltzer, ENIAC Programmer



“Programming is programming.

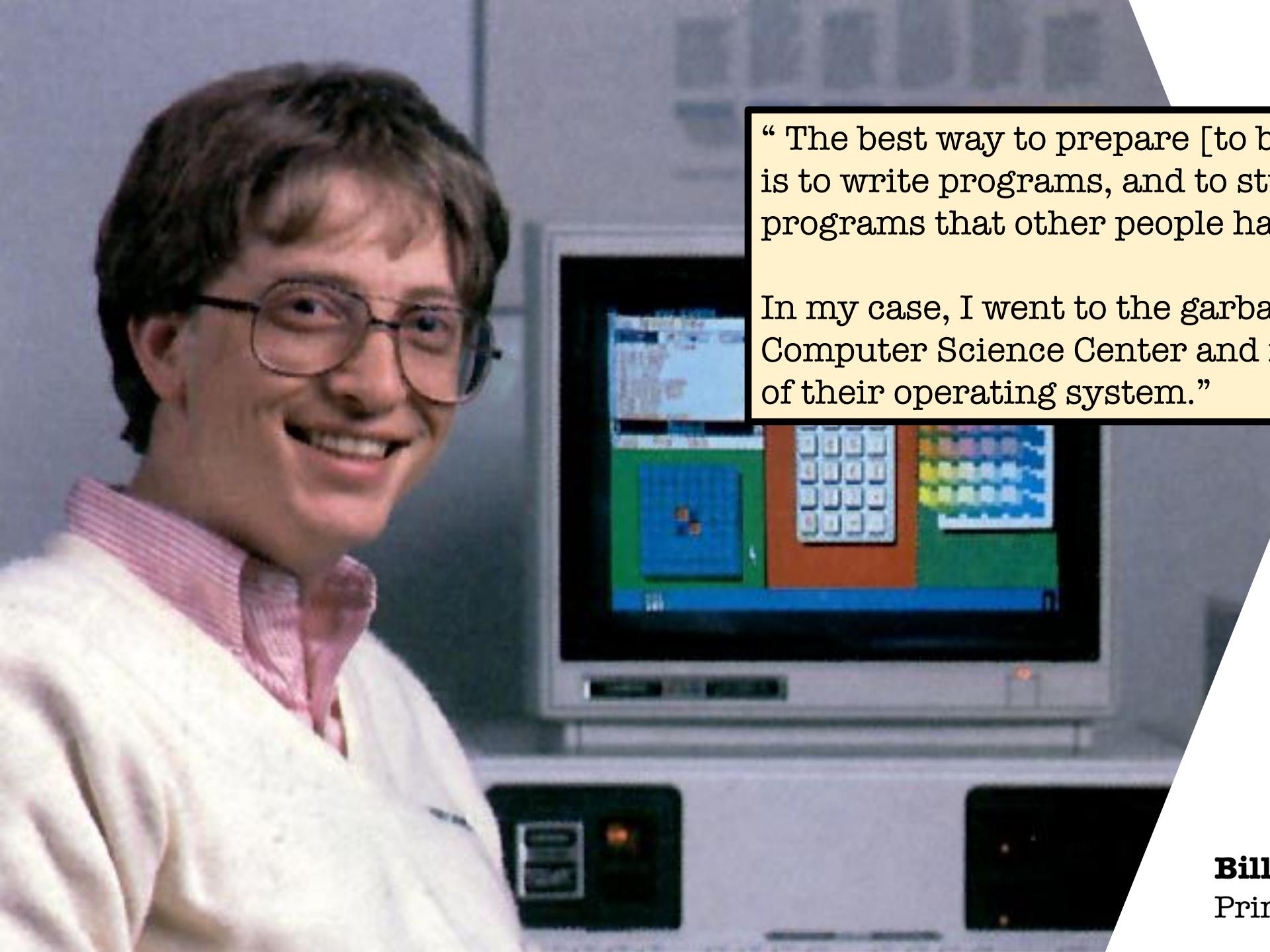
If you get good at programming, it doesn't matter which language you learned it in, because you'll be able to do programming in any language.

The hard part of programming is the same regardless of the language.

And if you have a talent for that, and you learned it here, you can take it over there.”

Richard Stallman

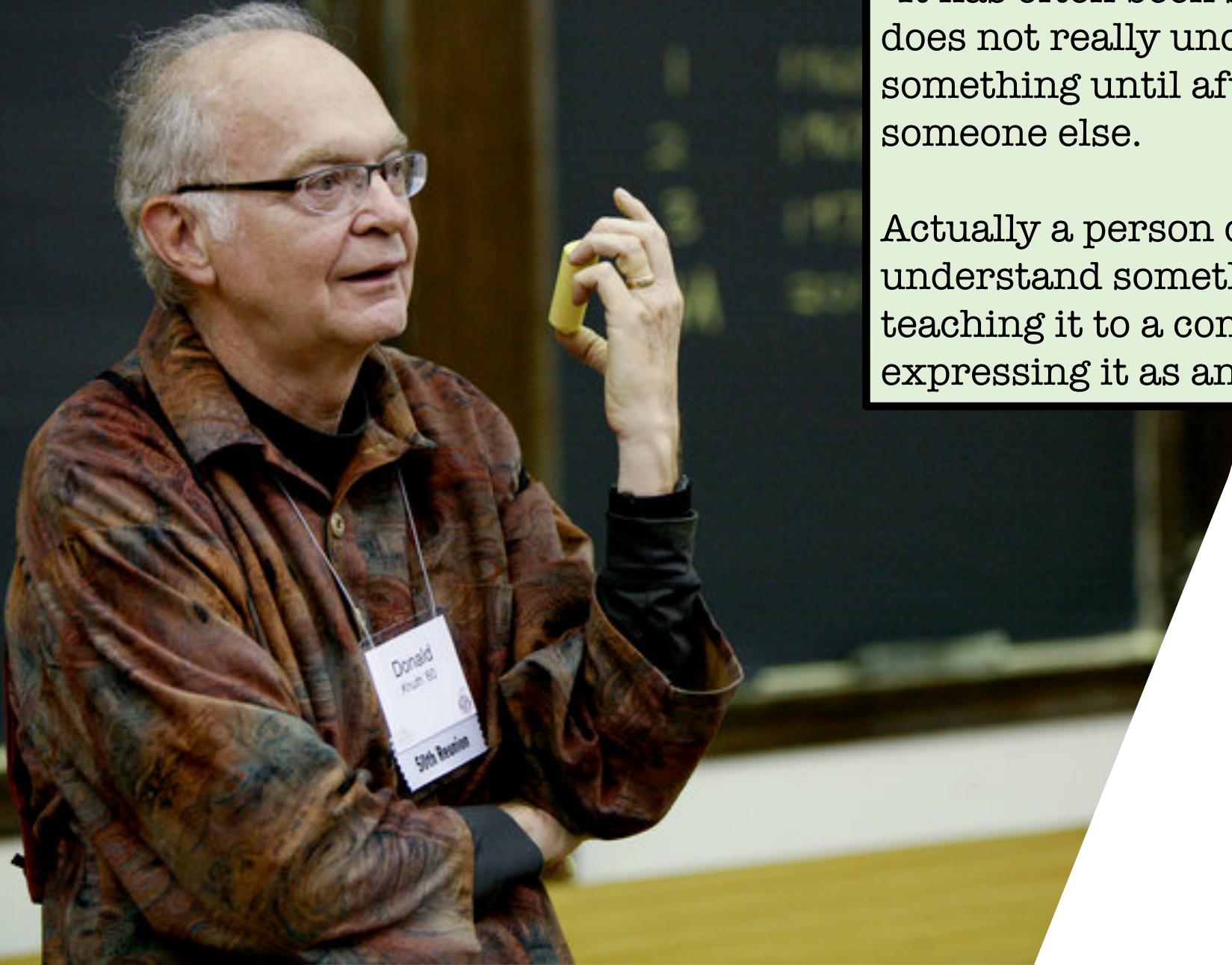
Founder of the Free-Software Movement



“ The best way to prepare [to be a programmer] is to write programs, and to study great programs that other people have written.

In my case, I went to the garbage cans at the Computer Science Center and fished out listings of their operating system.”

Bill Gates
Principal founder, Microsoft.



“It has often been said that a person does not really understand something until after teaching it to someone else.

Actually a person does not really understand something until after teaching it to a computer, i.e., expressing it as an algorithm.”

Don Knuth
1974 Turing Award Winner



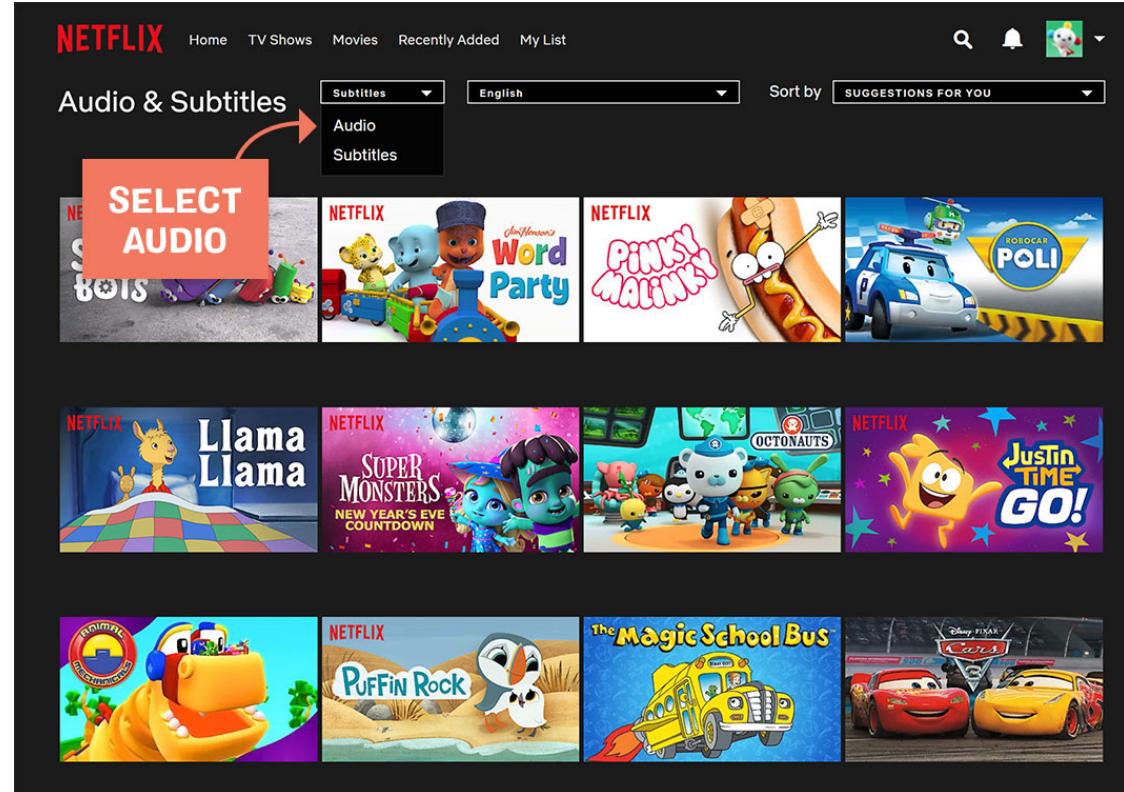
Data Structures

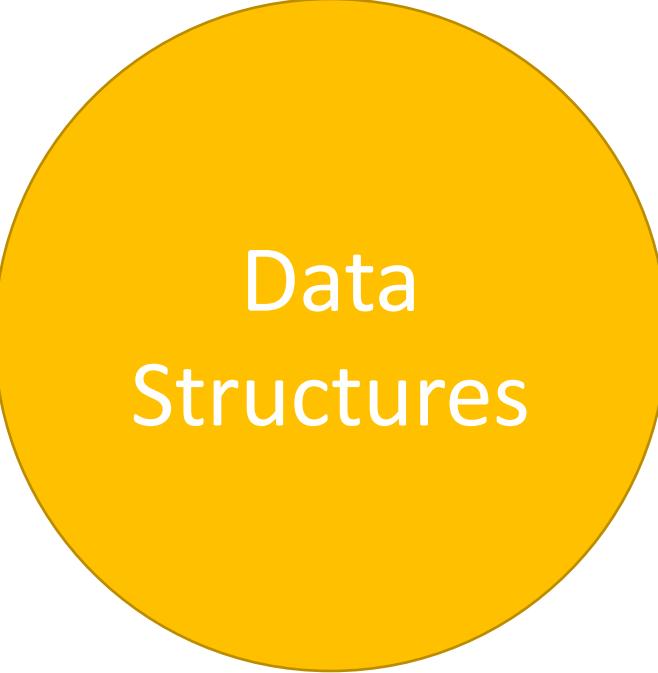
How to abstractly represent physical world to enable efficient problem solving?

Algorithms

Search Problem:

*Given a long and increasing sequence of key and value pairs,
find the value corresponding to a given key, if exists.*



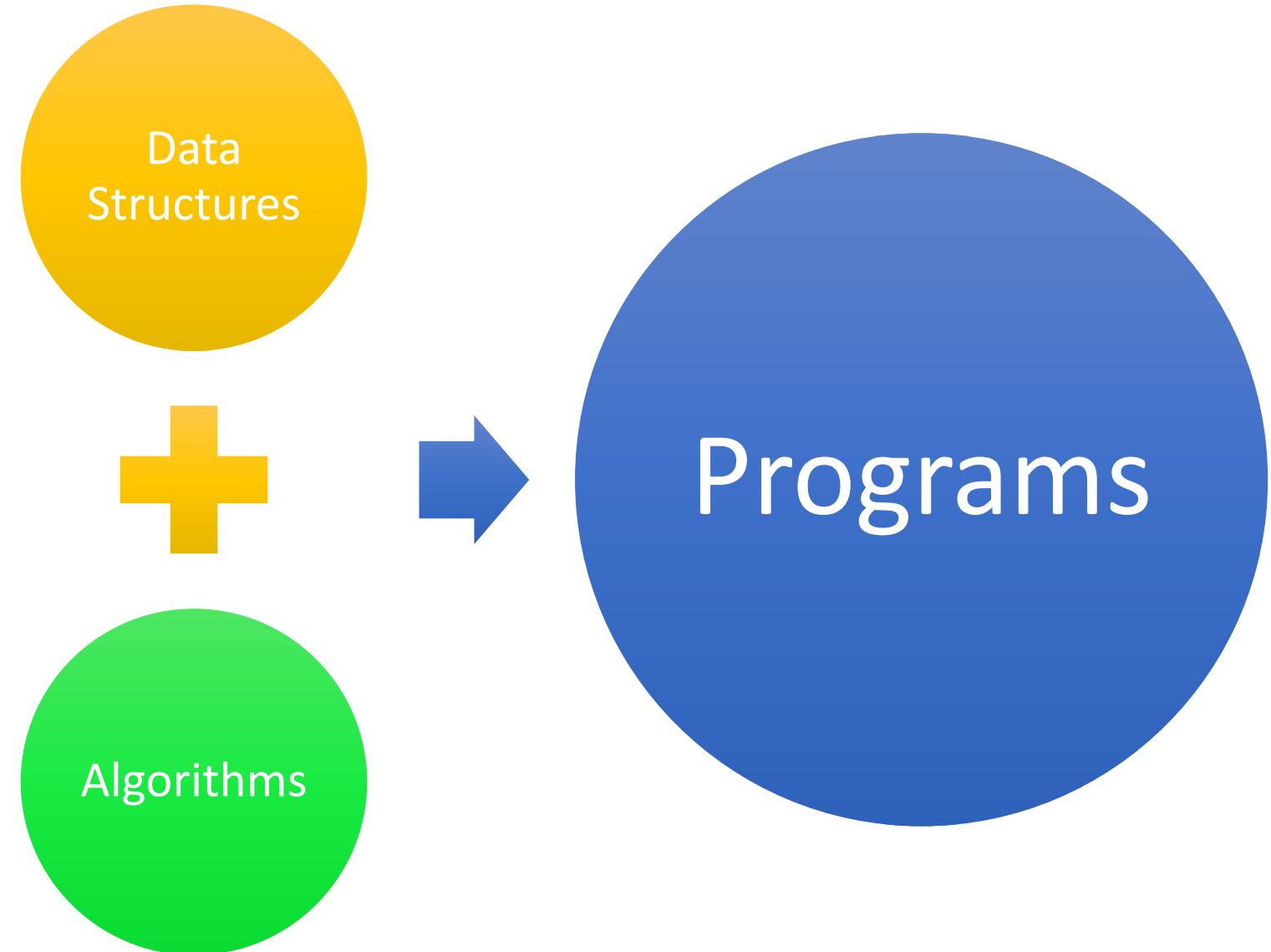


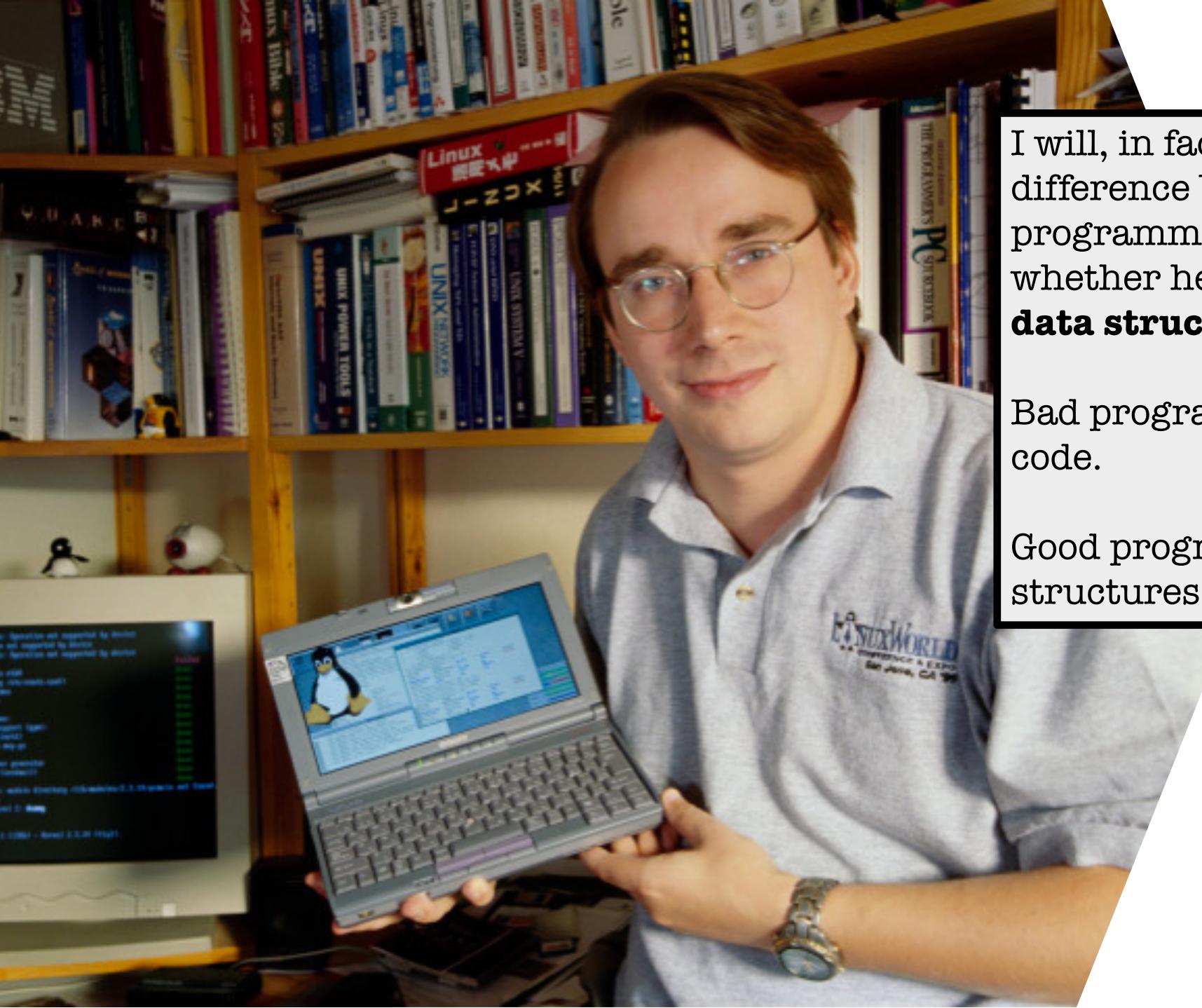
Data Structures

Abstract representation of problem data for computational problem-solving

Algorithms

Abstract Recipe to manipulate data
for computational problem solving





I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his **code** or his **data structures** more important.

Bad programmers worry about the code.

Good programmers worry about data structures and their relationships.”

Linus Torvalds
Creator and Principal Developer
of Linux kernel.

About Me

- Learned my data-structures in Pascal
- PhD in CS from the University of Warwick
- Assistant Professor of Computer Science at University of Colorado Boulder
- Research interests
 - Automata Theory, Logic, and Games
 - Formal methods for verification and synthesis
 - **AI Safety**
- Teaching Interests
 - **Algorithms and Data Structures**
 - **Theory of Computation**



A few well-established data-structures



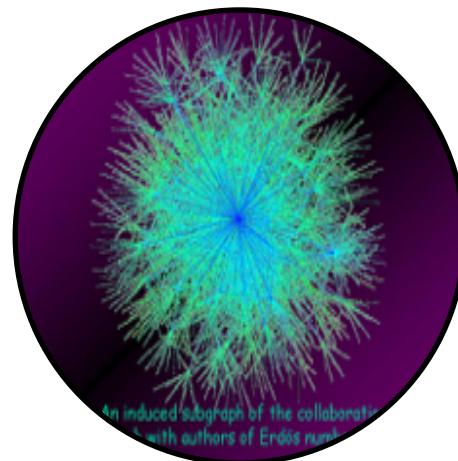
Stacks and Queues



Lists and HashMaps

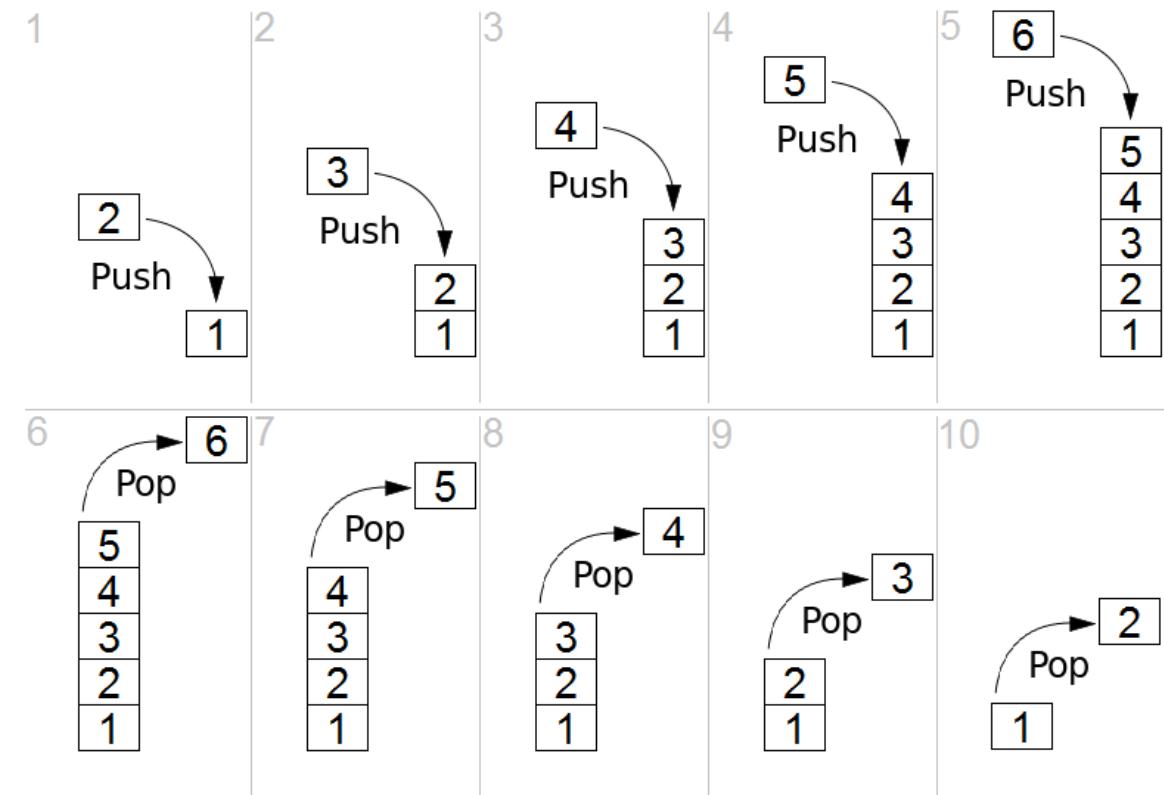
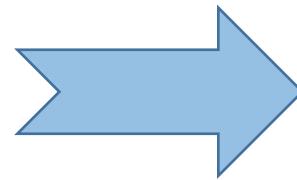
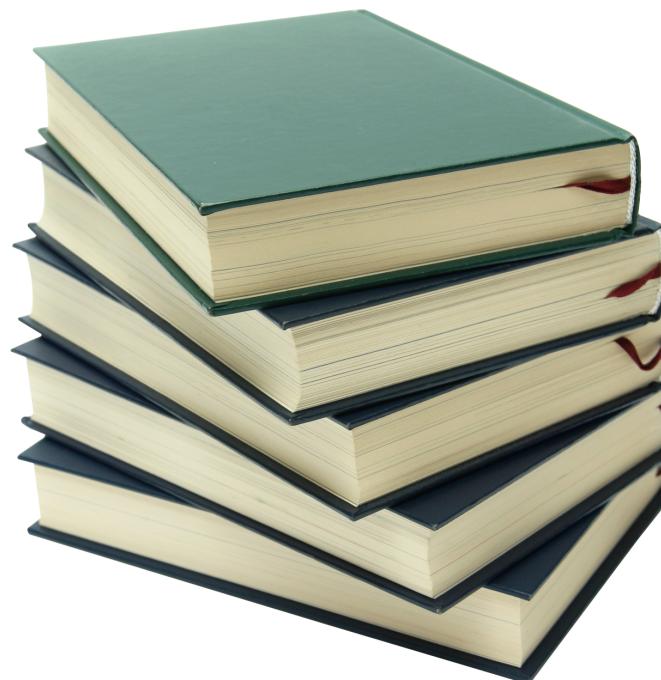


Trees

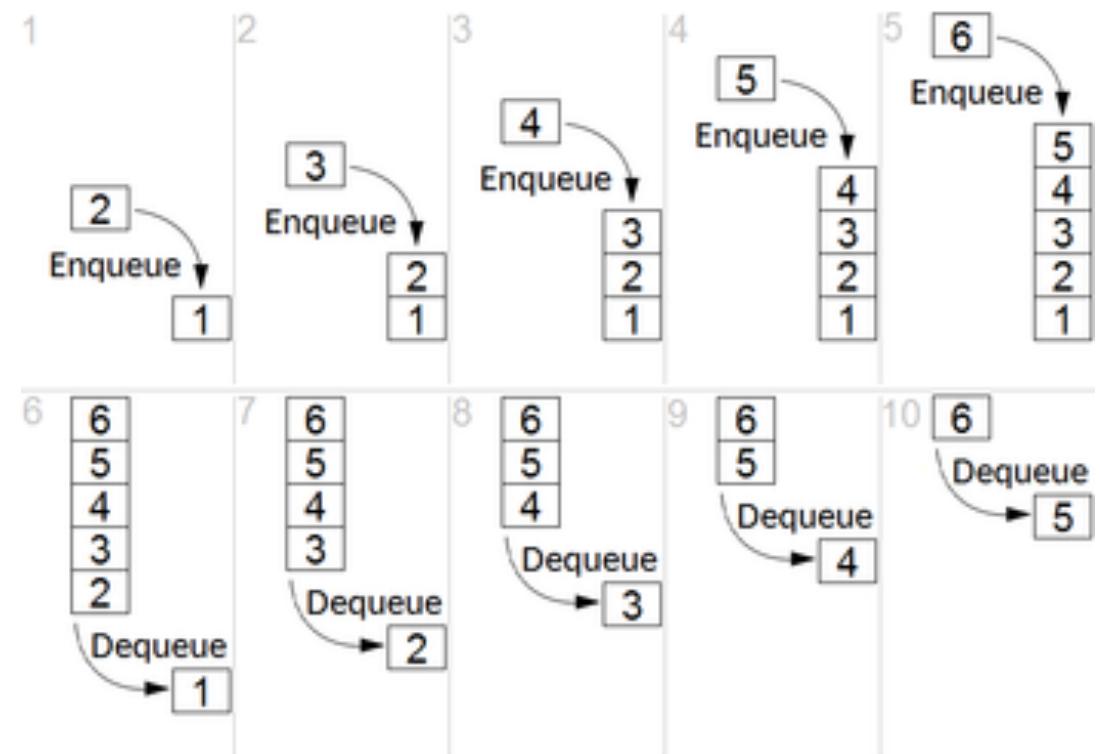
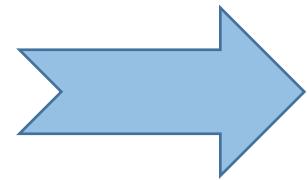
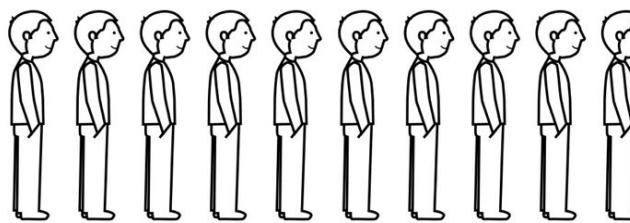


Graphs

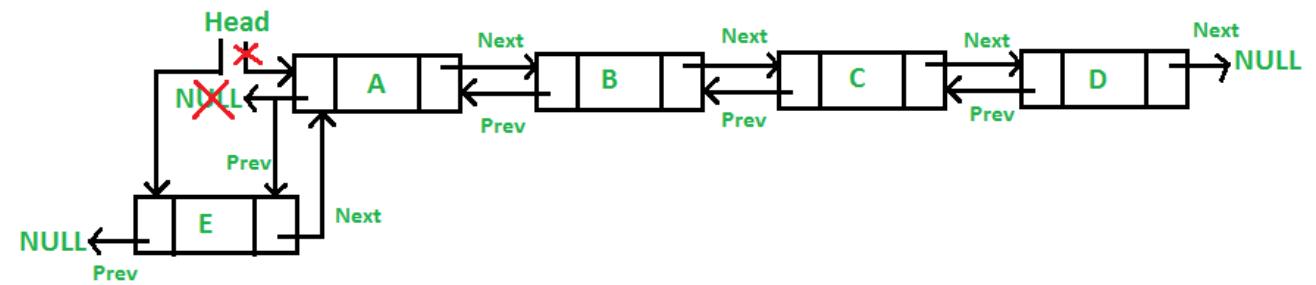
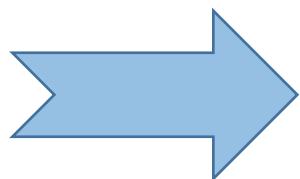
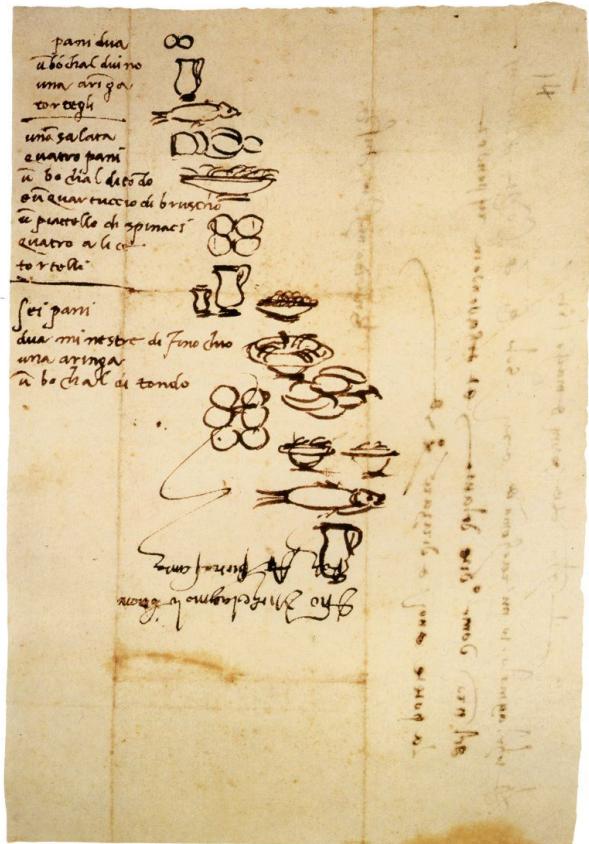
Examples of Data-Structures: Stacks



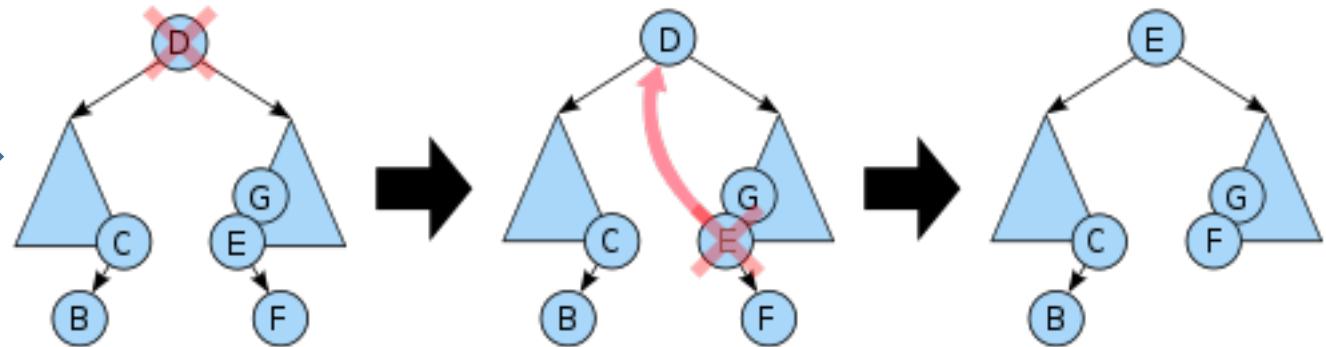
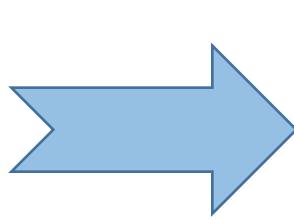
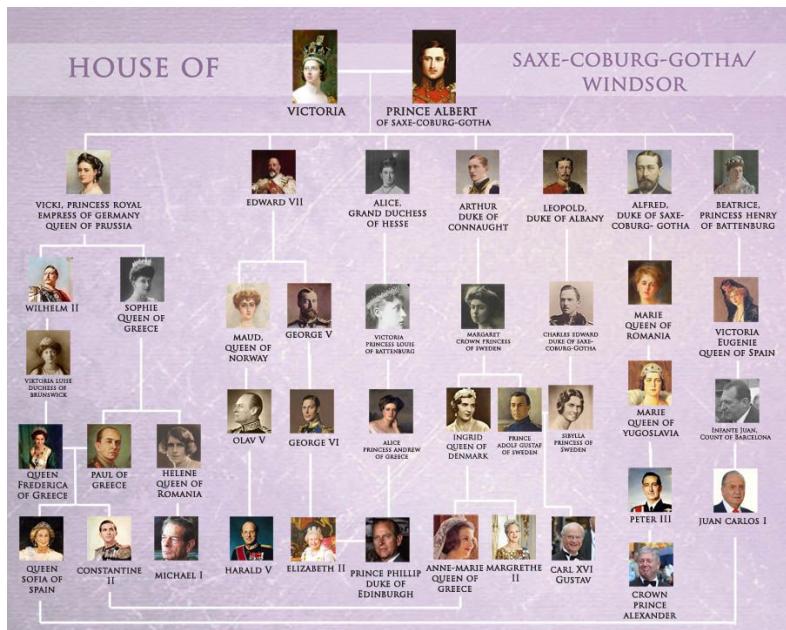
Examples of Data-Structures: Queues



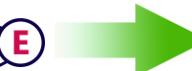
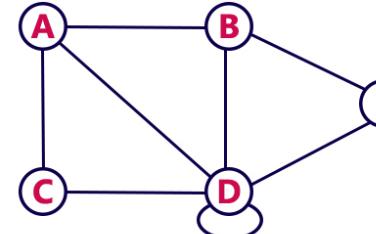
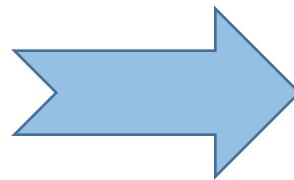
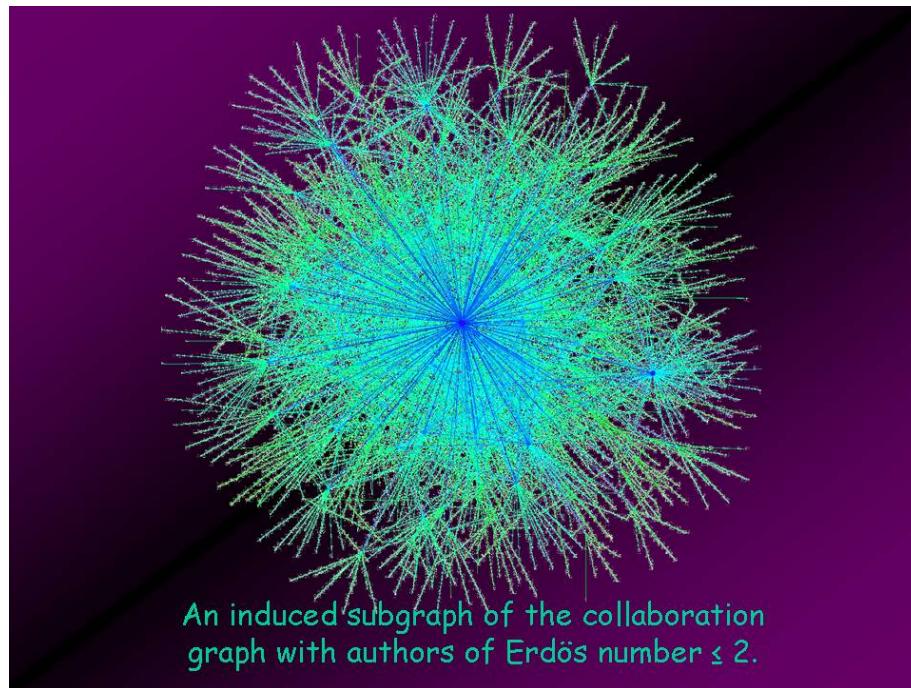
Examples of Data-Structures: Lists



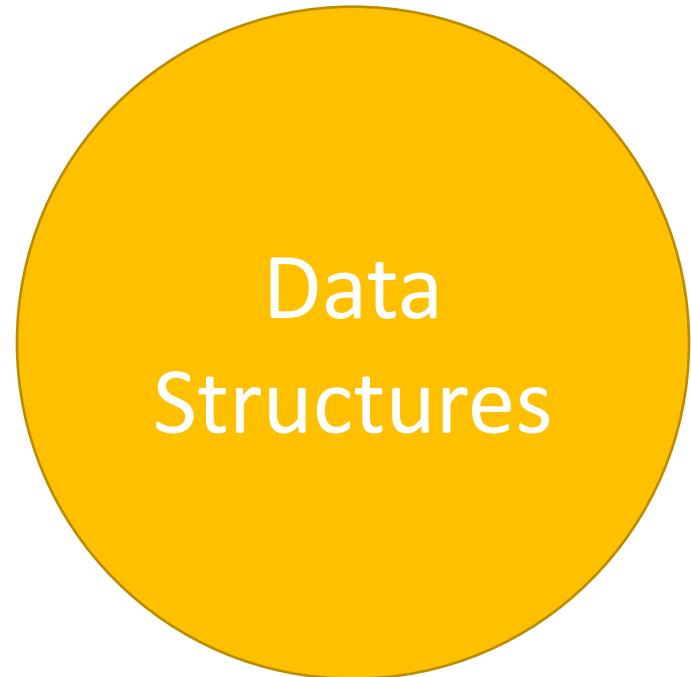
Examples of Data-Structures: Trees



Examples of Data-Structures: Graphs



$$\begin{array}{c} \begin{matrix} & \text{A} & \text{B} & \text{C} & \text{D} & \text{E} \\ \text{A} & \left[\begin{matrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{matrix} \right] \end{matrix} \\ \xrightarrow{\hspace{1cm}} \end{array}$$



In this course, you will:

1. study well-known data abstractions (e.g., stacks, queues, lists, trees) and their representation techniques (e.g., linking, arrays);
2. learn concepts used in algorithm design and analysis including criteria for selecting data structures to fit their applications.

Logistics

Your Section and Recitation Groups

- CSCI-2270-300
 - The other two sections are taught by **Maciej Zagrodzki** and **Asa Ashraf**
 - We essentially run it as one course
- Class meeting times
 - MWF 03:00pm—03:50pm MUEN E050
- Recitation
 - Weekly, 1.25-hour meetings.
 - Recitation activity. Ask questions about assignments and get extra help

Recitation Teaching Assistants

- Guohui Ding (guohui.ding@colorado.edu)
- Yu-Ju Lee (Yuju.Lee@colorado.edu)
- Saiyma (saiyma.sarmin@colorado.edu)
- Elizabeth Spaulding
(elizabeth.spaulding@colorado.edu)
- Varad Deshmukh
(varad.deshmukh@colorado.edu)
- Sharat Nimmagadda
(sharat.nimmagadda@colorado.edu)

Moodle

- **Course Website**
 - Syllabus, Announcements, Assignments
 - interview schedulers, lecture notes
 - grades (good idea to check that grade)
- **Enrollment steps**
 1. Go to: <http://moodle.cs.colorado.edu/>
 2. Login with your identikey and password
 3. Under Spring 2020, find course named:
CSCI 2270 - Zagrodzki, Ashraf, Trivedi - CS2: Data Structures
 4. Enrollment key: **DataStructs**



Office Hours

All instructors, TAs, and CAs will hold office hours throughout the semester.

1. A dynamic Google Calendar will hold all times and locations
2. Please come prepared with questions *after* putting thought into the given problem
3. Do not show up and expect to be walked through the assignment step-by-step

Syllabus

The syllabus is like a contract. Everyone is required to become familiar with it.

SYLLABUS

CSCI 2270: DATA STRUCTURES

SPRING 2020, 4 CREDITS, JANUARY 13 – MAY 6

Instructor and Lectures Information

Section 100

Instructor: Maciej Zagrodzki

Lecture: Monday, Wednesday, Friday, 10:00-10:50am; Location: VAC 1B20

Email: Maciej.Zagrodzki@colorado.edu

Office Location: ECOT 736

Section 200

Instructor: Asa Ashraf

Lecture: Monday, Wednesday, Friday, 2:00-2:50pm; Location: ECCR 1B40

Email: Asa.Ashraf@colorado.edu

Office Location: ECOT 538

Section 300

Instructor: Ashutosh Trivedi

Lecture: Monday, Wednesday, Friday, 3:00-3:50pm; Location: MUEN E050

Email: Ashutosh.Trivedi@colorado.edu

Office Location: ECCE 1B11

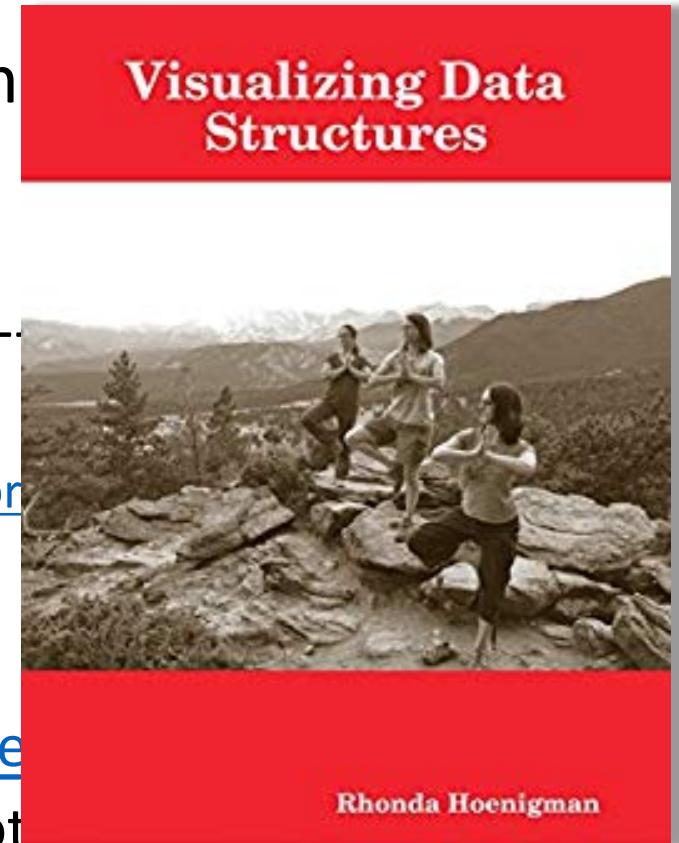
Recitations are a mandatory part of the course. Please check buffportal.colorado.edu for times and locations.

Piazza

- Online discussion forum where students can ask questions, answer questions, and explore the topics covered in class.
- **Not** a complaint board
- The forum is anonymous to other students, but it is **not** anonymous to the instructors.
- Inappropriate content will be removed, and violators will be banned.
- Students should not depend on getting last minute questions answered here. We cannot require our course staff to work weekends!

Logistics (Contd.)

- Prerequisite (One of the following with C- minimum)
 - CSCI 1300/1310/1320 or ECEN 1030/1310
 - APPM 1345/1350 or MATH 1300/1310
 - Proficient in a Programming Language (preferably C++)
 - Resources for learning C++
 - Online C++ Textbook <http://www.cplusplus.com/files/tutor/>
 - C++ videoTutorials (links available from Moodle)
- Textbook
 - [VDS] Hoenigman, R. 2015. [Visualizing Data Structures](#)
 - The VDS eBook, Course materials, such as lecture notes, assignments, quizzes will be made available in electronic form on the Moodle.



Logistics (Contd.)

- Introduce fundamentals of data-structures
- Introduce pseudo-code and implementations
- Bring your laptop as well as pencil-and-paper
- Weekly 75-minute recitation with course-TAs
- Reinforces the ideas discussed in class
- Opportunity to ask questions, get clarification on lectures/notes
- Weekly programming assignments to apply concepts
- Problem sets assignment weekly, due the following week
- Lecture quizzes (online questions) at the end of every week to refresh the concepts

Lectures

Recitations

Assignments

Quizzes

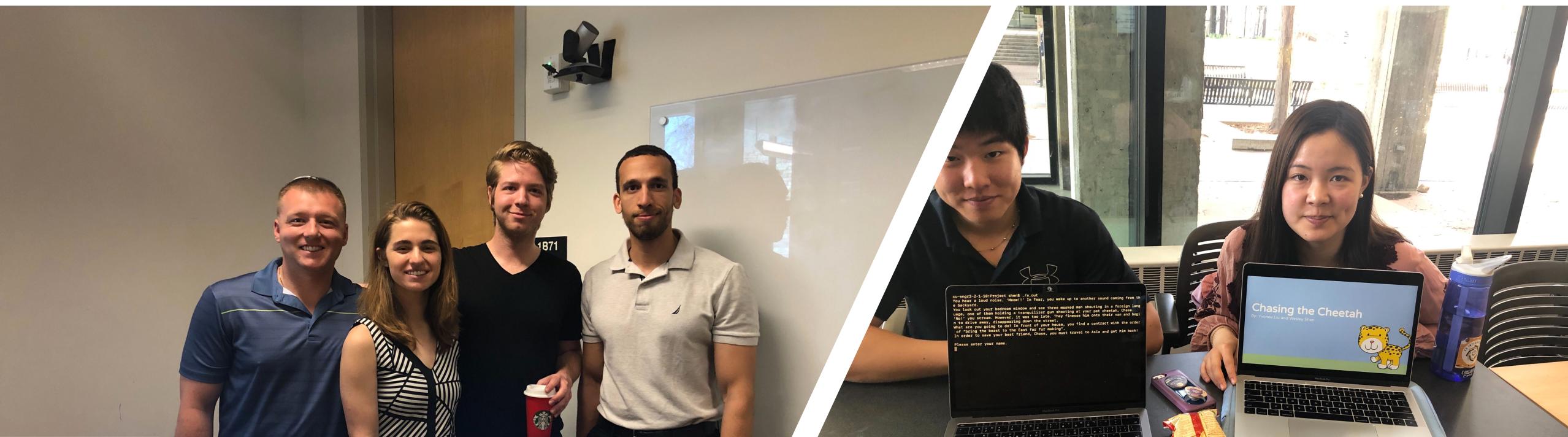
Logistics (Contd.)

- Two midterms (evening exams)
- Must have 65% average on the exams to get better than a D+ in the class, regardless of other grades
- A grade of C- in this class is required to take the next class in the Computer Science sequence
- The optional Final Exam can be used to replace your lowest midterm grade.
- In the final project you get an opportunity to demonstrate your overall understanding of data-structure and your ability to apply them in real world scenarios.

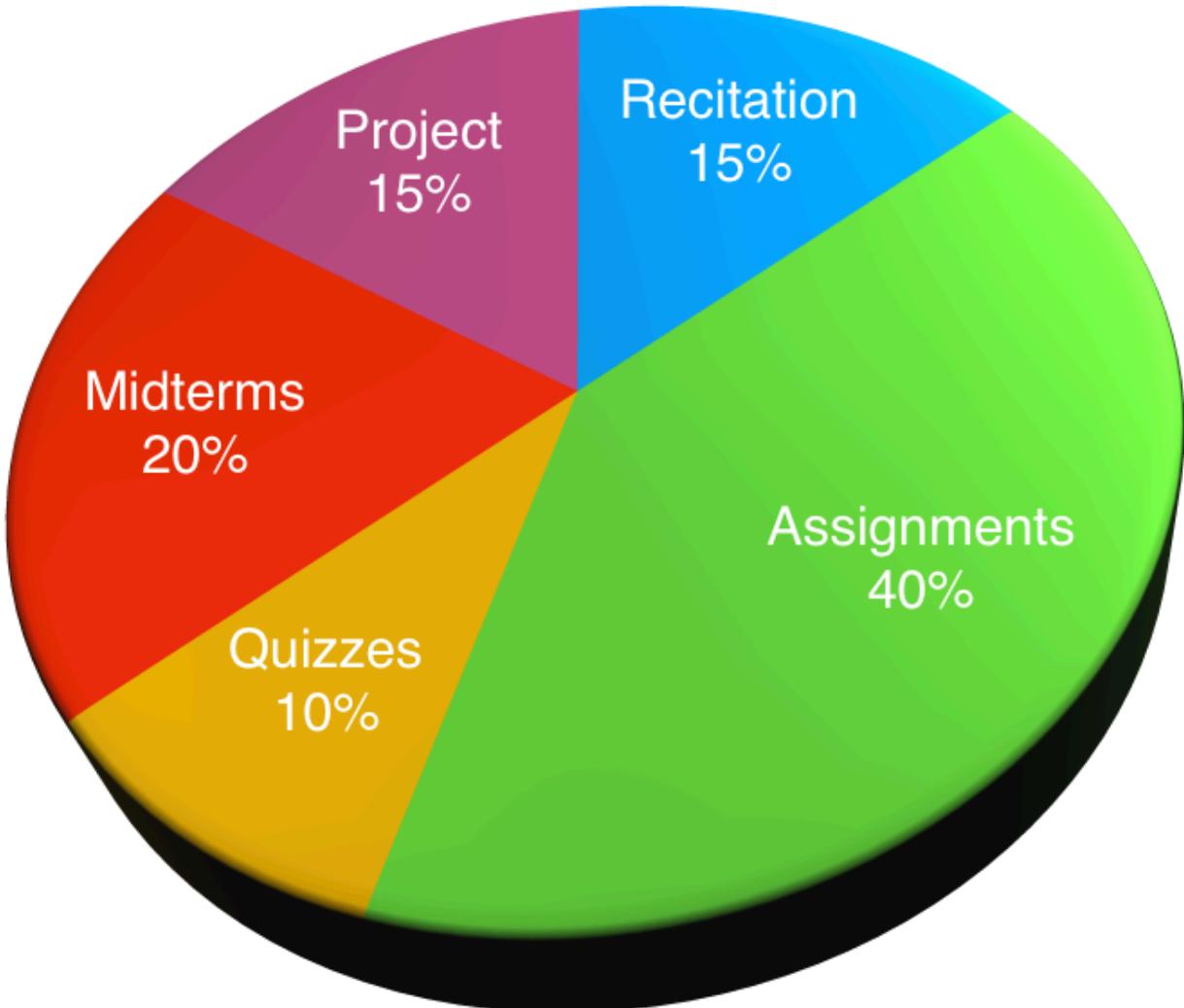
Midterms

Final
Exam

Final
Project



Grading



Recitation this week

- Get prepared for the semester:
 - Polish your skills with your favorite text editor:
 - [Emacs](#), [Vim](#), [Atom](#), [Sublime](#),
 - [Visual Studio Code](#)
 - Install a [C++ compiler](#) on your machine.
 - Please read the syllabus carefully (on Moodle) with special focus on classroom behavior and collaboration policies.
- Create a **Moodle** account (Enrollment key: **DataStructs**).
- Create a **Piazza** account.
- Recitations are graded, so please attend them.