

CS50





CS50









Stelios

THE
NORTH
FACE



[OPINION](#) [UNIVERSITY](#) [CITY](#) [SPORTS](#) [SCITECH](#) [CULTURE](#) [YTV](#) [WKND](#) [MAG](#) [PHOTO](#) | [ABOUT](#) [SUBSCRIBE](#)



CS50 TAs learn to teach their peers

STEPHANIE ROGERS | SEP 24, 2015

STAFF REPORTER

When Mary Farner '16 prepared for her first CS50 section, she was nervous. She had never before taught a Yale course section in front of her peers — now her students.





[OPINION](#) [UNIVERSITY](#) [CITY](#) [SPORTS](#) [SCITECH](#) [CULTURE](#) [YTV](#) [WKND](#) [MAG](#) [PHOTO](#) | [ABOUT](#) [SUBSCRIBE](#)



ULAs expand to other CS courses

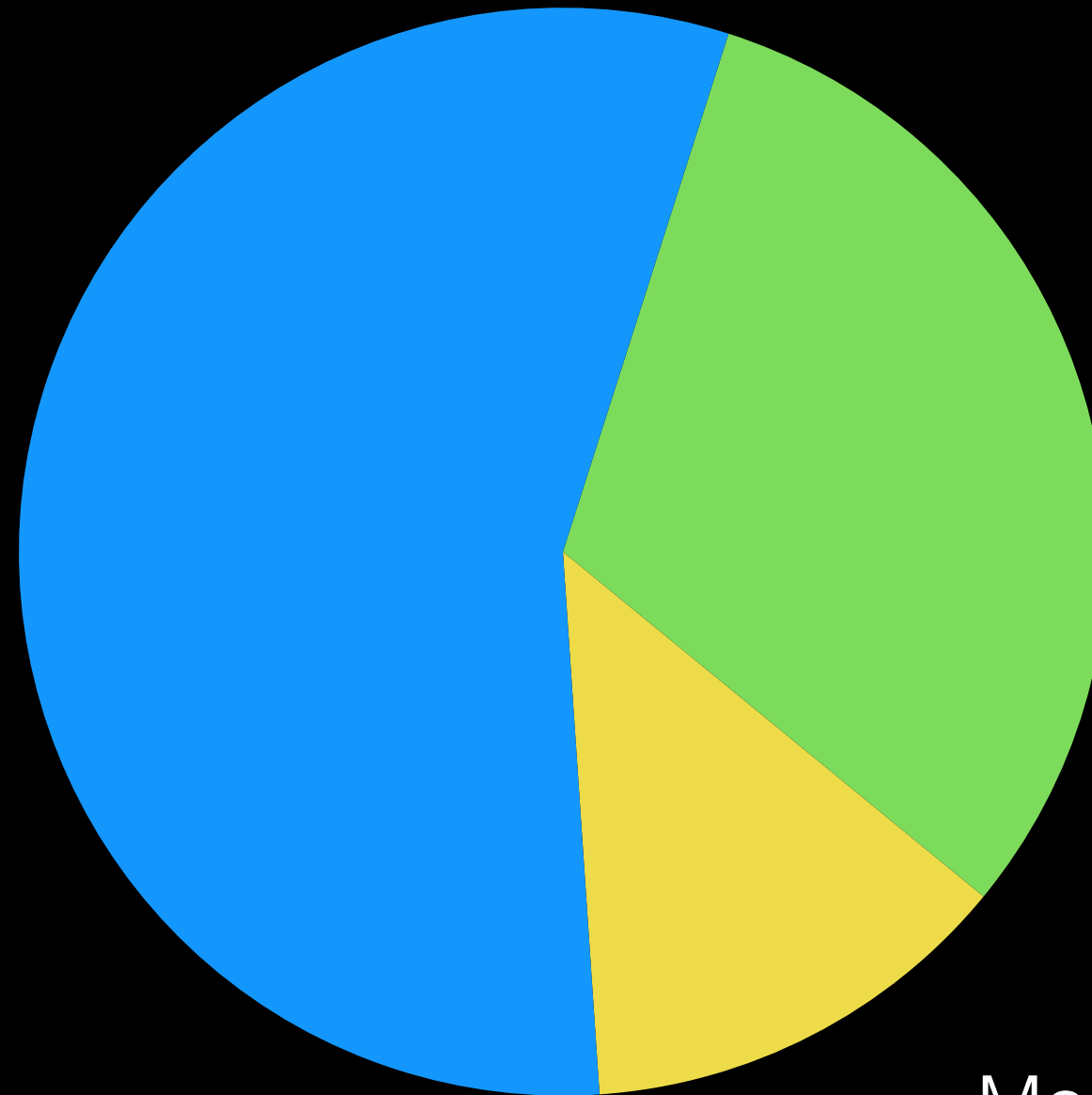
DANIELA BRIGHENTI & VICTOR WANG | APR 08,
2016

STAFF REPORTERS



what ultimately matters in this course is not so much where you end up relative to your classmates but where you, in **Week 11**, end up relative to yourself in **Week 0**

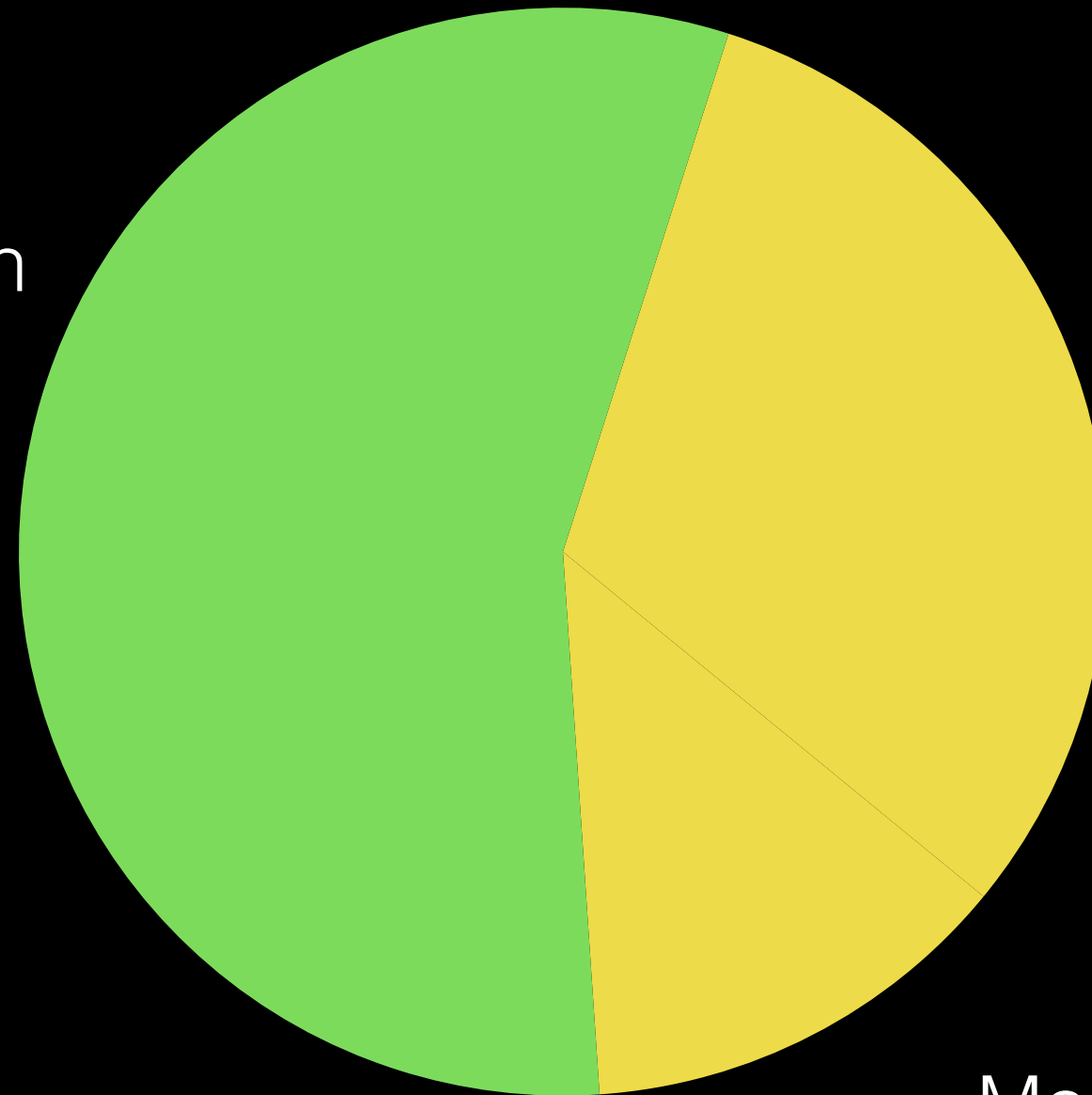
Less Comfortable



Somewhere in Between

More Comfortable

Somewhere in Between



More Comfortable

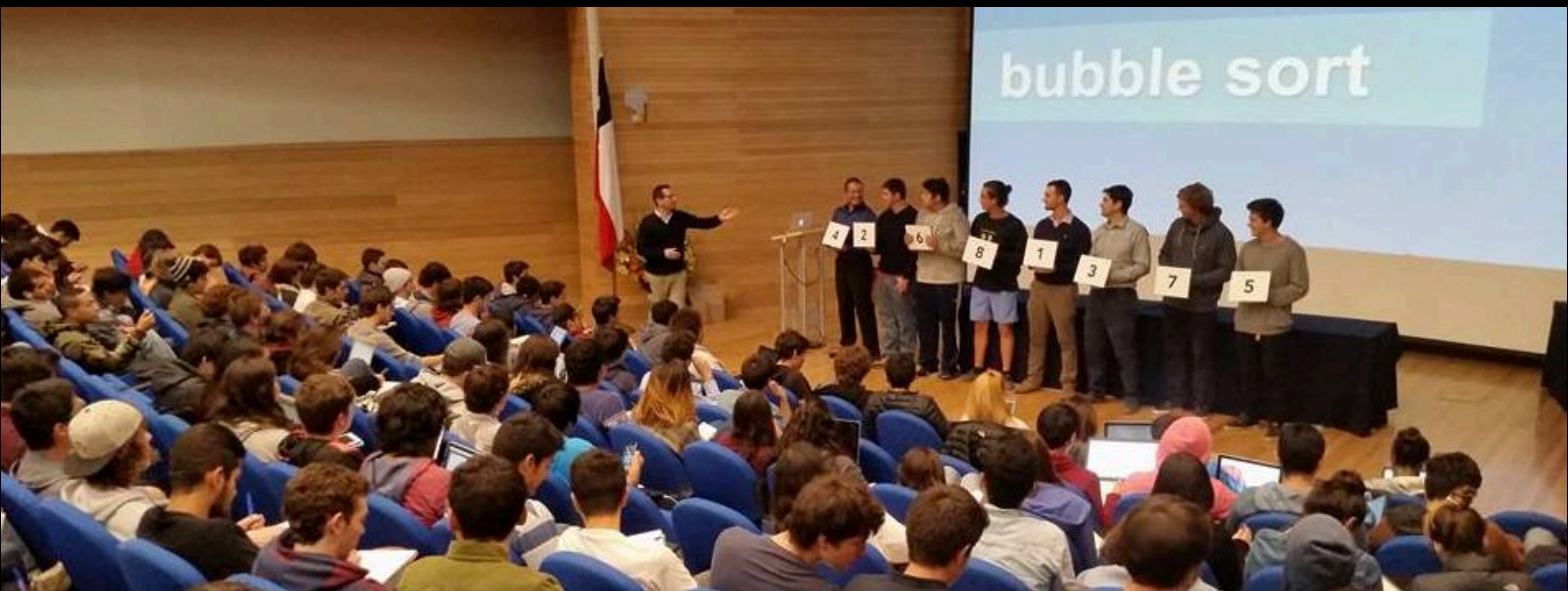
cs50.yale.edu/apply



CS50 Teachers around the world ★







bubble sort

4

2

6

8

1

3

7

5











This is CS50 UTEPSA 2016



This is CS50 *Nicaragua*

Fundación Uno



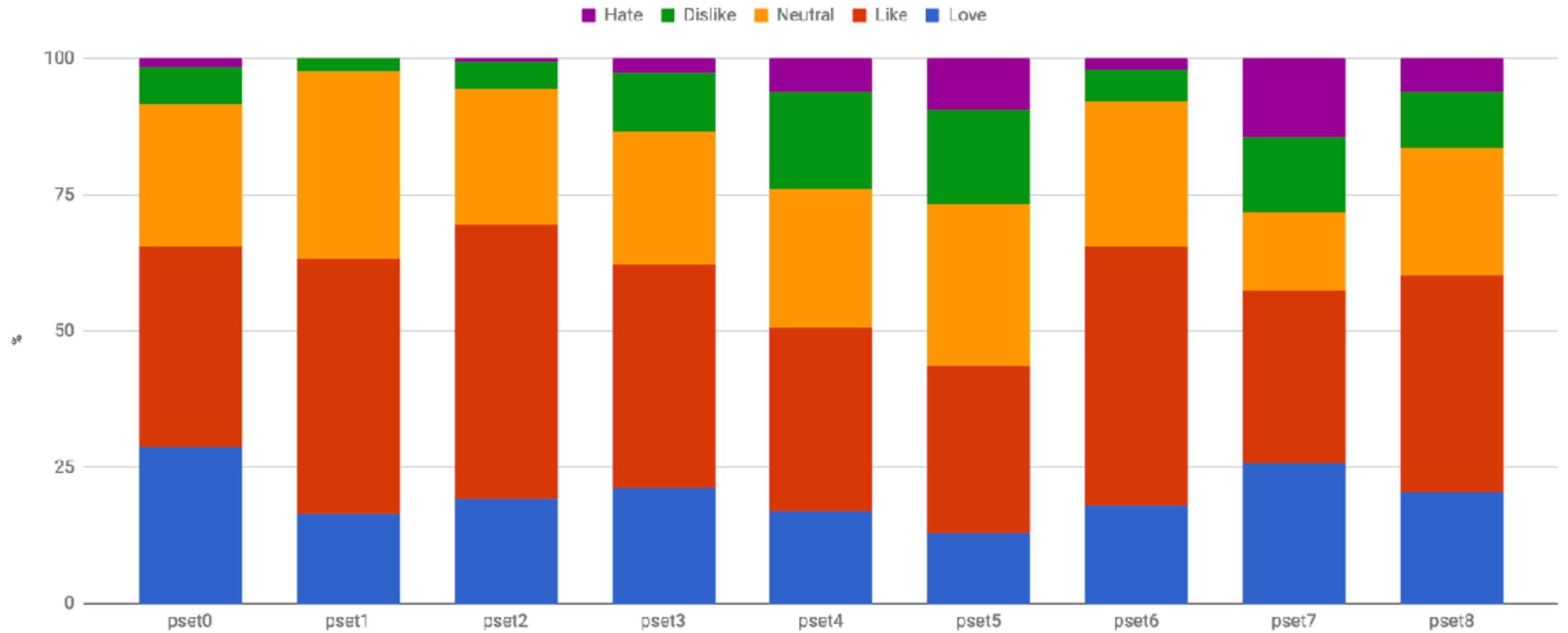




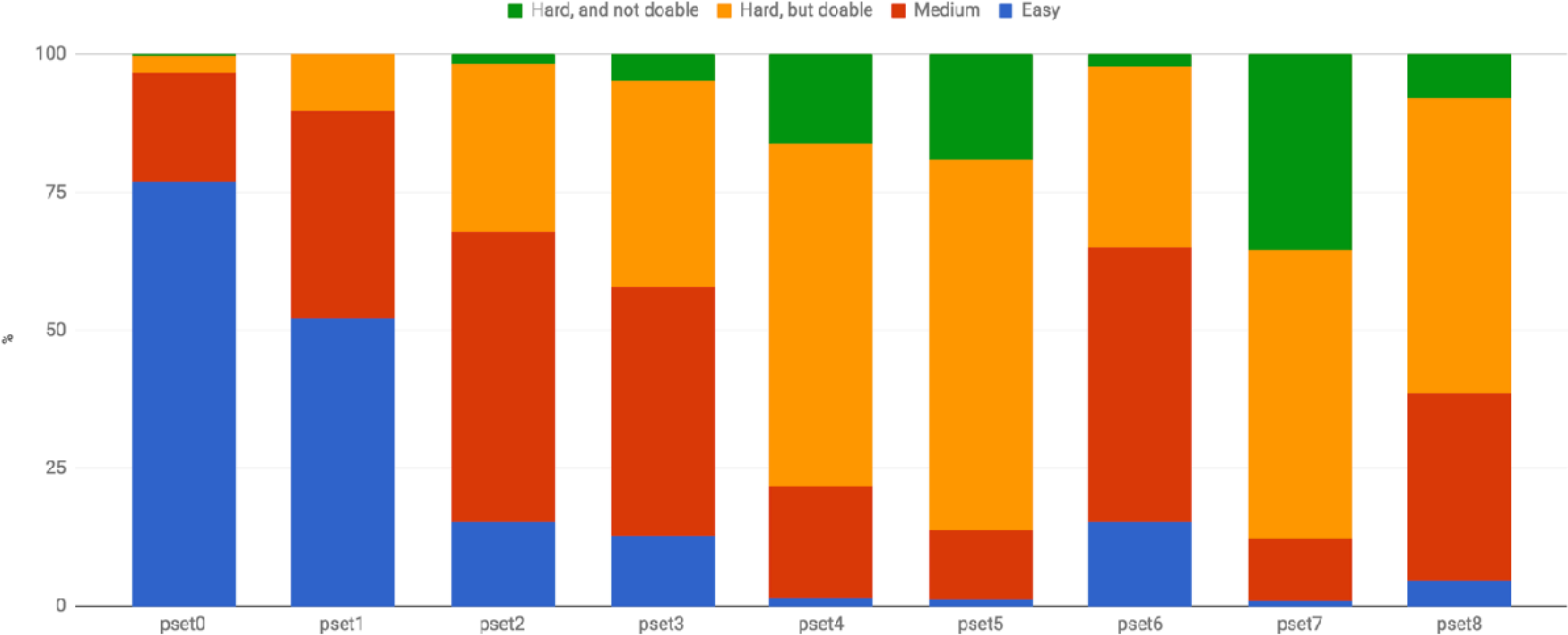
cs50.yale.edu/apply

problem solving

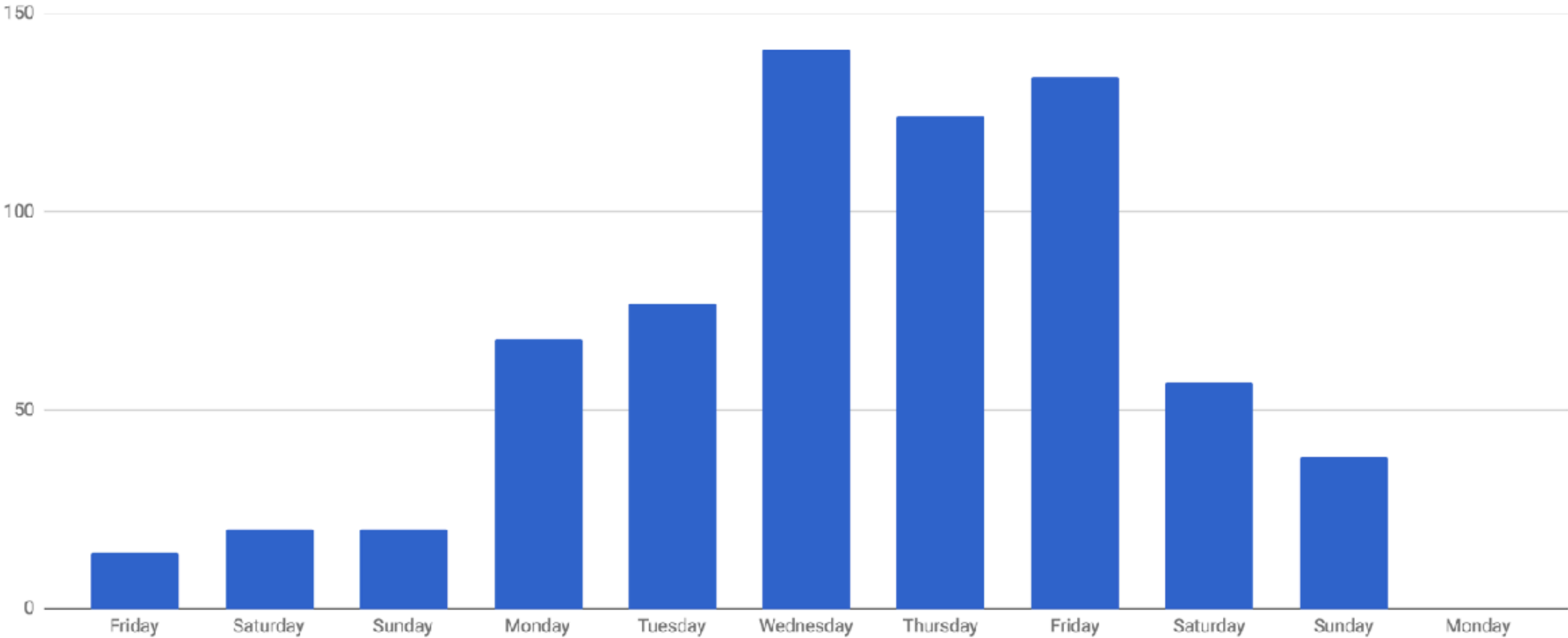
What did you think of problem sets?



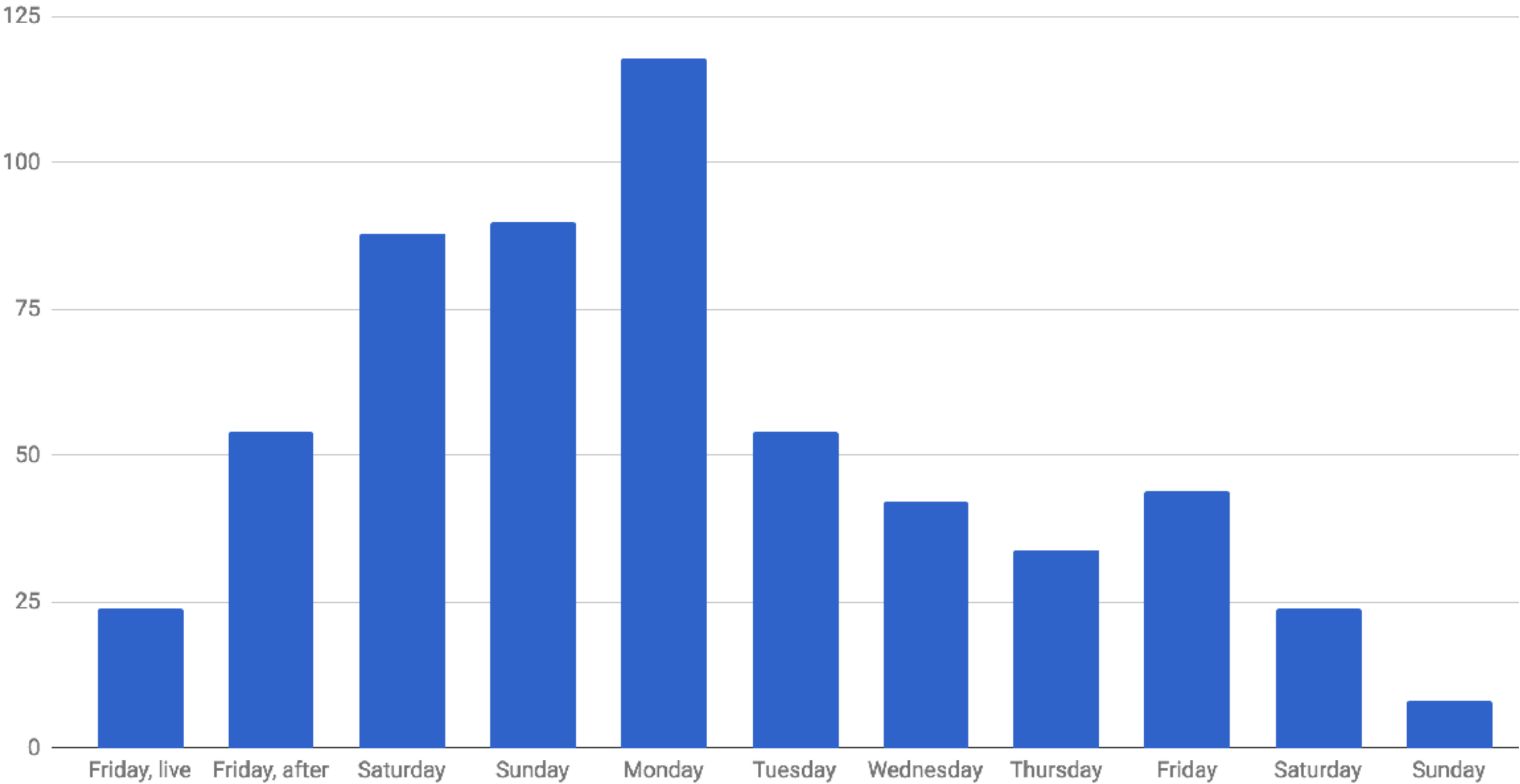
How difficult were problem sets?



When did you start problem sets?



When did you watch lectures?



problem solving

Level

1

Death

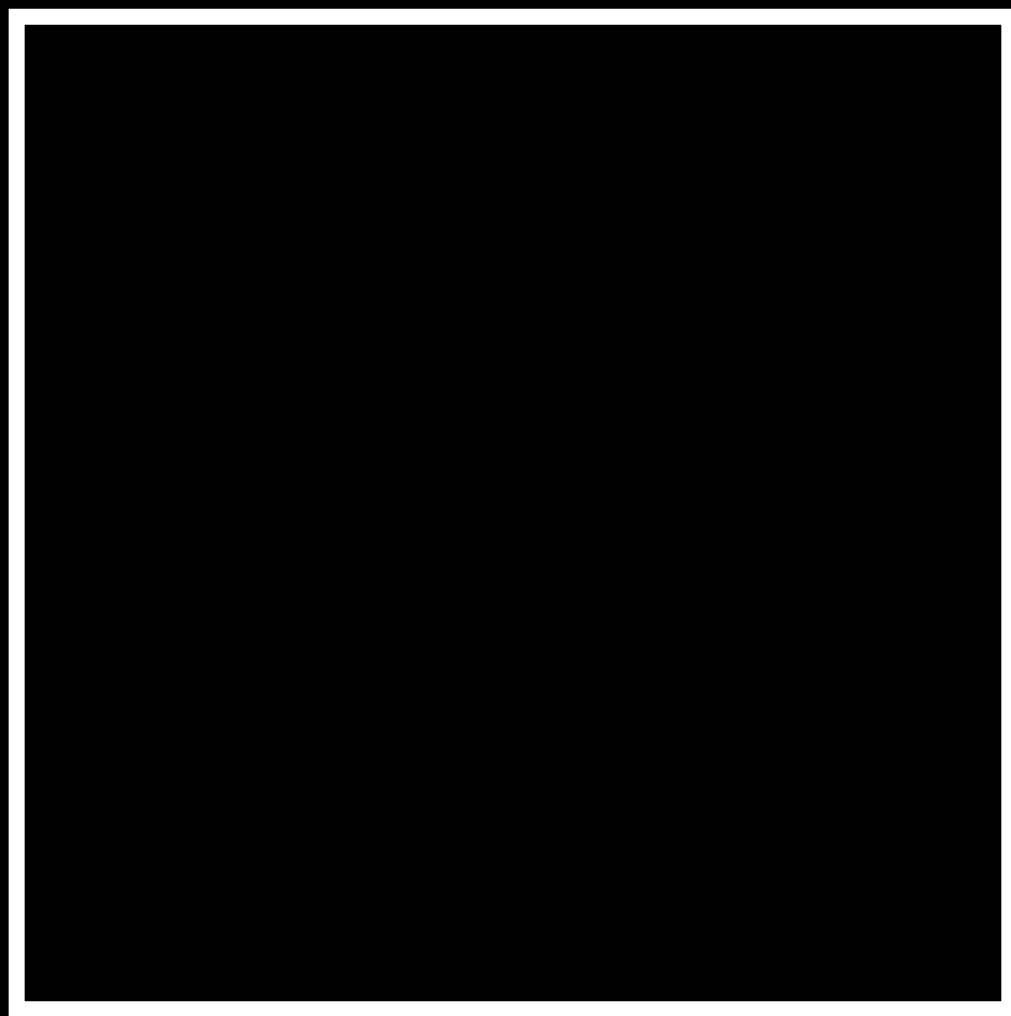
0

Arrow Keys to
Move

Y



inputs →



→ outputs

algorithms

Life after CS50

Learn git

https://youtu.be/MJUJ4wbFm_A

Git?

people.

the original.



Sign up for free private repos

https://education.github.com/discount_requests/new

Which best describes you? 

☒ **Student**

☐ **Teacher**

☐ **Researcher**

☐ **Administrator/staff**

☐ **Other**

What are you looking to get a discount for? 

☒ **Individual account**

☐ **Organization account**

Next

Install git on your own Mac or PC

<https://help.github.com/articles/set-up-git/>

Try Sublime Text

<https://www.sublimetext.com/>

FOLDERS

- ▼ tensorflow
 - ▶ tensorflow
 - ▶ third_party
 - ▶ tools
 - ▶ util
 - .gitignore
 - ACKNOWLEDGMENTS
 - <> ADOPTERS.md
 - AUTHORS
 - /* BUILD
 - CODEOWNERS
 - <> CONTRIBUTING.md
 - <> ISSUE_TEMPLATE.md
 - LICENSE
 - <> README.md
 - <> RELEASE.md
 - WORKSPACE
 - configure
 - /* models.BUILD
- ▼ sqlite3
 - /* shell.c
 - /* sqlite3.c
 - /* sqlite3.h
 - /* sqlite3ext.h

base64.cc

```
34
35 void base64_encode(const uint8_t * data, size_t len, char * dst,
36                   base64_charset variant)
37 {
38     const char * charset = (variant == base64_charset::URL_SAFE)
39         ? URL_SAFE_CHARSET
40         : STANDARD_CHARSET;
41
42     size_t src_idx = 0;
43     size_t dst_idx = 0;
44     while (src_idx < len)
45     {
46         uint8_t s0 = data[src_idx];
47         uint8_t s1 = data[src_idx + 1];
48         uint8_t s2 = data[src_idx + 2];
49
50         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
51         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
52         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
53         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
54     }
55
56     if (src_idx < len)
57     {
58         uint8_t s0 = data[src_idx];
59         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
60
61         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
62         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
63         if (src_idx + 1 < len)
64             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
65     }
66
67     dst[dst_idx] = '\0';
68 }
69
```

Use Multiple Selections to rename variables quickly

Try Atom

<https://atom.io/>

- atom
 - .git
 - .github
 - apm
 - benchmarks
 - docs
 - dot-atom
 - electron
 - exports
 - keymaps
 - menus
 - nade_modules
 - out
 - resources
 - script
 - spec
 - src

```
272
273   getComponent () {
274     if (!this.component) {
275       this.component = new TextEditorComponent({
276         element: this,
277         mini: this.hasAttribute('mini'),
278         updatedSynchronously: this.updatedSynchronously
279       })
280       this.updateModelFromAttributes()
281     }
282
283     return this.component
284   }
285 }
286
287 module.exports =
288 document.registerElement('atom-text-editor', {
289   prototype: TextEditorElement.prototype
290 })
291
```


Try vim

<http://valloric.github.io/YouCompleteMe/>

```

int LongestCommonSubsequenceLength( const std::string &first,
                                     const std::string &second ) {
    const std::string &longer = first.size() > second.size() ? first : second;
    const std::string &shorter = first.size() > second.size() ? second : first;

    int longer_len = longer.size();
    int shorter_len = shorter.size();

    std::vector<int> previous( shorter_len + 1, 0 );
    std::vector<int> current( shorter_len + 1, 0 );

    for ( int i = 0; i < longer_len; ++i ) {
        for ( int j = 0; j < shorter_len; ++j ) {
            if ( toupper( longer[ i ] ) == toupper( shorter[ j ] ) )
                current[ j + 1 ] = previous[ j ] + 1;
            else
                current[ j + 1 ] = std::max( current[ j ], previous[ j + 1 ] );
        }

        for ( int j = 0; j < shorter_len; ++j ) {
            previous[ j + 1 ] = current[ j + 1 ];
        }
    }

    return current[ shorter_len ];
}

```


Host a web app

<https://cs50.io/>

Share this workspace



Links to share

Editor:

<https://ide.cs50.io/username/ide50>

☐ Public

Application:

<https://ide50-username.cs50.io>

☒ Public

Files:

<https://preview.cs50.io/username/ide50>

☐ Public

Who has access

▼ Read+Write

 You

RW

☐ Don't allow members to save their tab state

Invite People

username or email

Invite

☒ Notify people via email

☐ R ☒ RW

Done

Share this workspace



Links to share

Editor:

<https://ide.cs50.io/username/ide50>

Application:

<https://ide50-username.cs50.io>

Files:

<https://preview.cs50.io/username/ide50>



Public

Who has access

▼ Read+Write

● You

RW



Don't allow members to save their tab state

Invite People

username or email

Invite



Notify people via email

R

RW

Done

Share this workspace



Links to share

Editor:

<https://ide50-username.cs50.io>

☐ Public

Application:

<https://ide50-username.cs50.io>

☒ Public

Files:

<https://preview.cs50.io/username/ide50>

☐ Public

Who has access

▼ Read+Write

● You

RW

☐ Don't allow members to save their tab state

Invite People

username or email

Invite

☒ Notify people via email

☐ R ☒ RW

Done

Host a web app

<https://www.heroku.com/platform>

Host a web app

<http://awseducate.com/>

Get a domain

<https://nc.me/>

Get student developer pack

<https://education.github.com/pack>



CS50 Hackathon



LET'S GO
BULLDOGS

1517

denver

VA



LET'S GO
BULLDOGS























WELCOME

Want IHOP favorites?
call ahead we'll
have it ready to go
617 787-0533

ihop
'n go
CARRY OUT

CS50 Fair













CSO
Fall
Emmi Datta '18

CSO
Fall
Andi Peng '18

CSO
Fall
Kafina Shiu '18

CSO
Fall
Assistant
Fall '18







Small informational plaque next to the deer exhibit.







CS50



Yale CS Courses

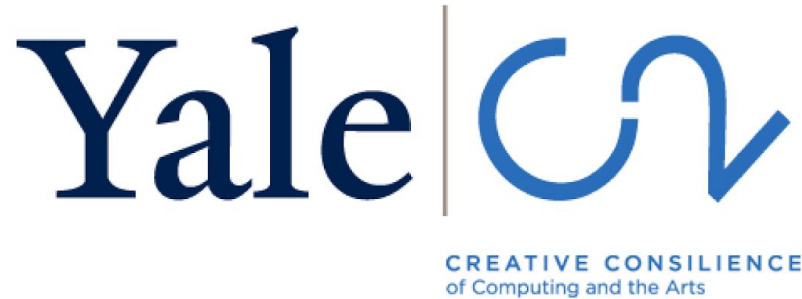
You are ready for these CS courses this Spring

- CS 035: 21st Century Computer Music (Freshman Seminar)
 - CS 078: See it, Change it, Make it (Freshman Seminar)
 - CS 200: Intro to Information Systems
 - CS 201: Intro to Computer Science (Prereq for most upper-level classes)
 - CS 213: Apps, Software, and Entrepreneurship
 - CS 276: Digital Humanities Apps
-
- CS 223: Data Structures and Programming Techniques (after taking 201)

We have various CS and CS + X majors

- B.A., B.S., B.A./M.A. Computer Science (talk to Prof. Aspnes)
 - CS + Math (talk to Prof. Aspnes)
 - CS + Electrical Engineering (talk to Prof. Aspnes)
 - CS + Psychology (talk to Prof. Aspnes)
-
- Computing and the Arts (talk to Prof. Dorsey)
 - CS + Architecture, Art, History of Art, Music, Theater

Computing & the Arts



What is Computing and the Arts?

An interdepartmental major designed for students interested in integrating work in computing and one of the arts disciplines. Computing and the Arts provides students with core Computer Science skills that allow them to explore interesting and substantive problems in architecture, art, art history, music, or theater.

Computer Science Core Studies

6 courses, including
CPSC 201, 202, and 223



Architecture

6 courses from
ARCH

Art

6 courses
from ART

Art History

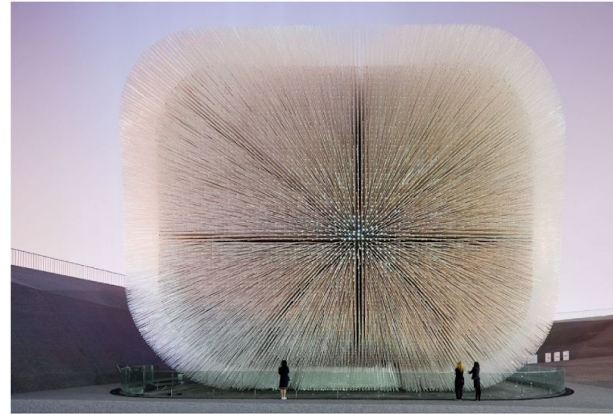
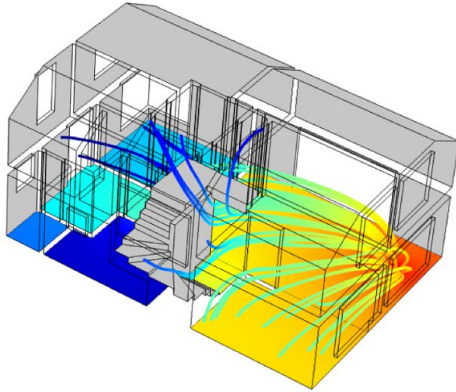
6 courses, 5 from
HSAR and one
studio art course

Music

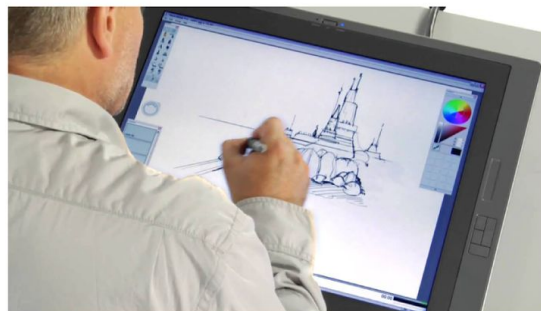
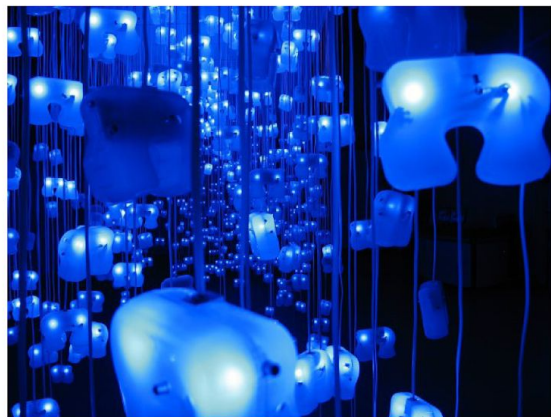
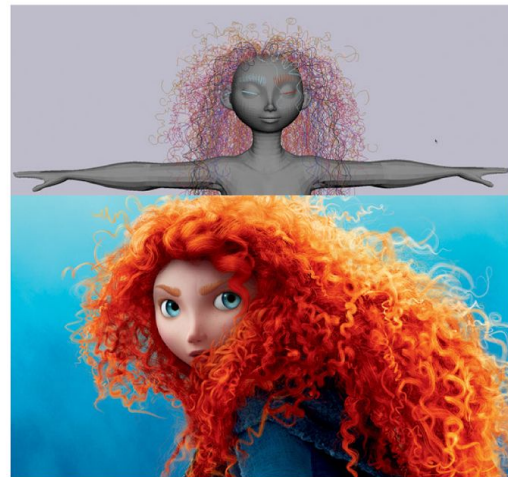
6 courses
from MUSI

Theater Studies

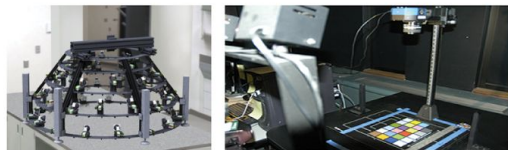
6 courses, 3
from THST
and 3 courses
in dramatic
literature or
theater history



Architecture



Art



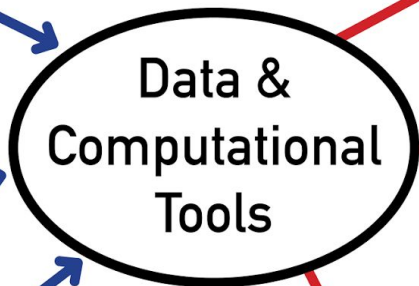
Imaging



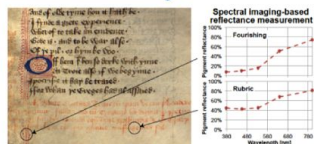
Conservation



3D Scanning



Humanities



Databases

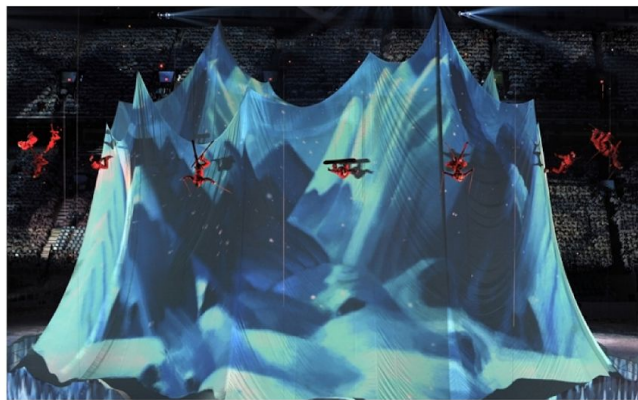
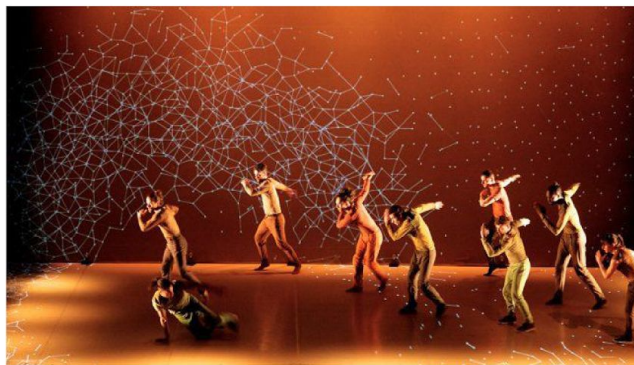
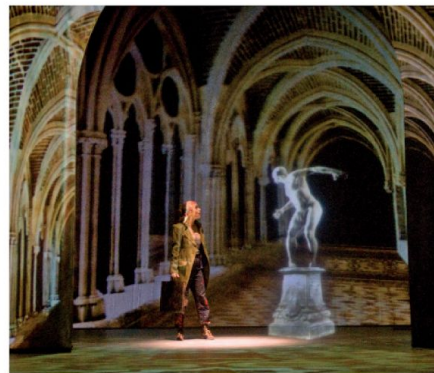
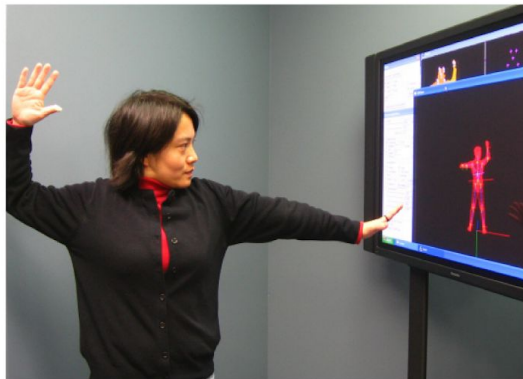


Communication

Art History



Music



Theater Studies

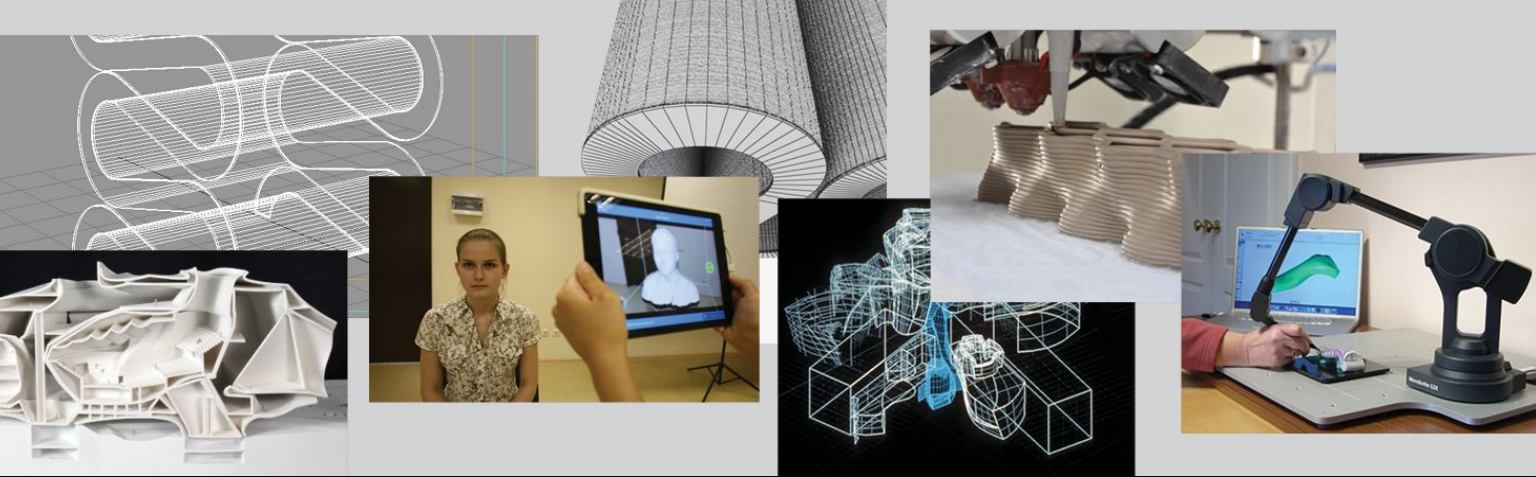


**COOPER
HEWITT**



Career Opportunities

For more information about the
Computing and the Arts Major, please contact
DUS Julie Dorsey at julie.dorsey@yale.edu



ARCH 009/CPSC 078

See it, Change it, Make it

Spring 2018
Prof. Julie Dorsey

Description

This course provides a hands-on introduction to the **theory** and **practice** of digital capture, modeling and fabrication. Topics include digital representations of shape, **3D scanning**, shape **modeling** and **editing**, and physical **production**, including 3D printing, milling and laser cutting. Develops an understanding of these capabilities through exercises and modeling-and-fabrication projects. Architectural forms at a variety of scales will be used as a vehicle for exploration and experimentation.

Prereqs and Enrollment

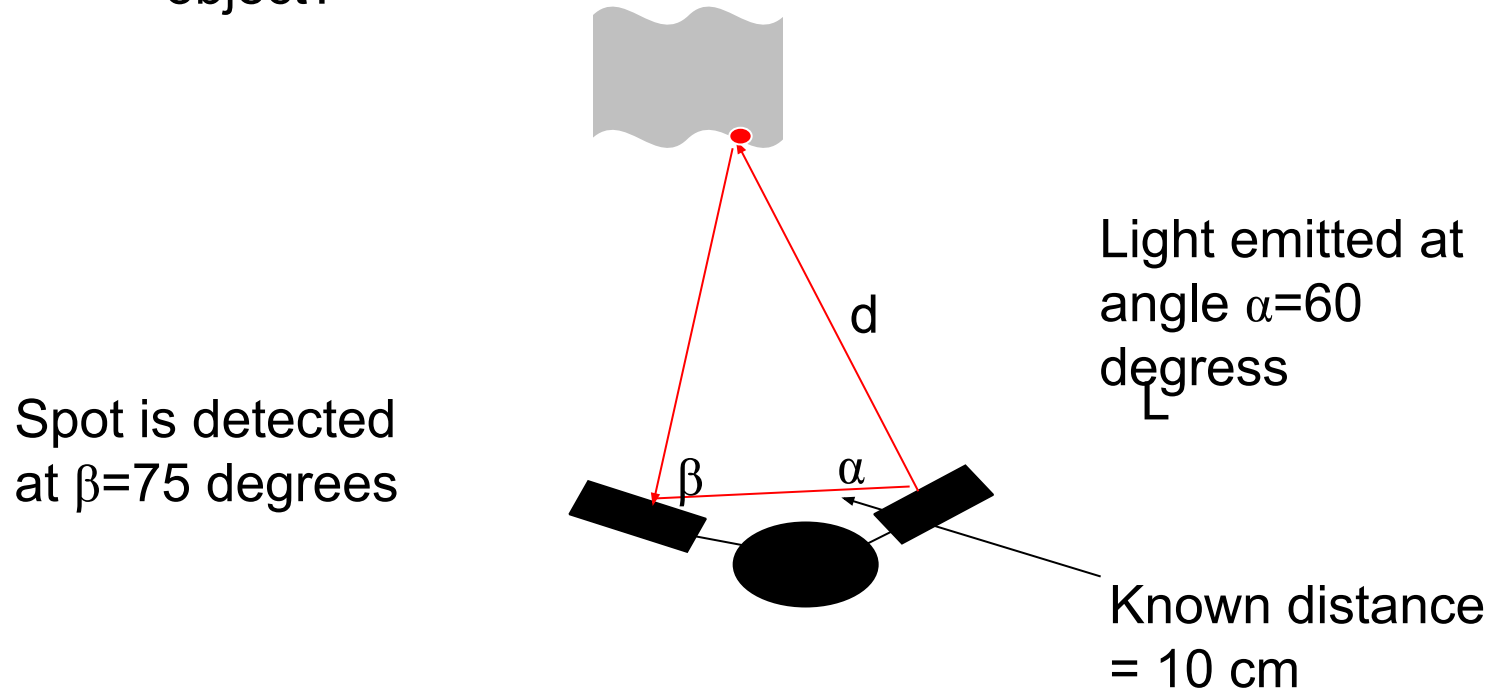
No course prerequisites. Students are expected to be proficient in high school level algebra, trigonometry, and geometry.

Limited to freshman. No exceptions.

Enrollment is limited. You must enter the freshman-seminar lottery in order to be eligible to take the course.

Sample Problem: Triangulation Scanner Principle:

What is the distance d from the laser to the object?



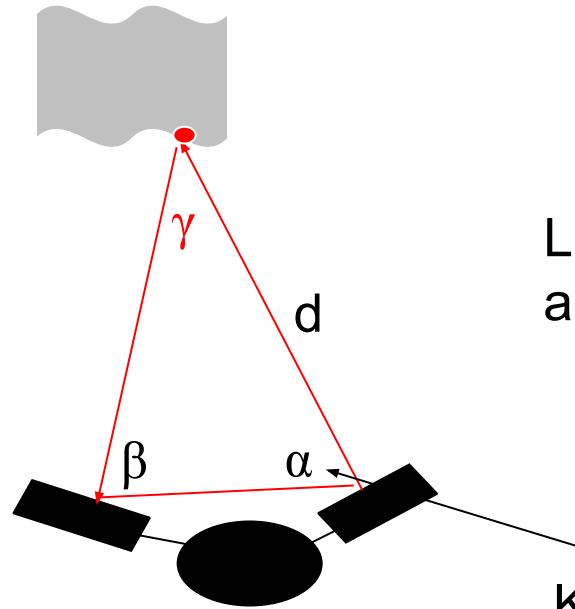
2. Triangulation Scanner Principle:

What is the distance d from the laser to the object?

$$\gamma = 180 - \beta - \alpha = 45^\circ$$

By the law of sines,
 $\sin(\gamma)/10 = \sin(\beta)/d$
 $d = 10 * \sin(\beta) / \sin(\gamma)$
 $d \approx 13.7 \text{ cm}$

Spot is detected
at $\beta = 75^\circ$



Light emitted at
angle $\alpha = 60^\circ$

Known distance
= 10 cm

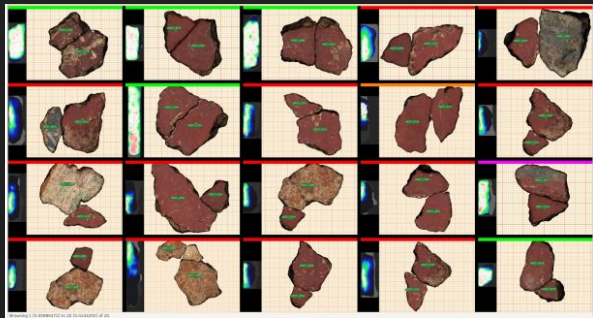
How the course will operate

- Lecture-lab-discussion format
- One or two guest lectures
- Field trip to Blue Sky Studios
(Greenwich, CT)

CS 276: Applications in the Digital Humanities

CS 276: Applications in the Digital Humanities

- Learn to develop web applications (à la pset 8)
- Discuss applications in art, archeology, music, literature
 - Detect and analyze meter in poems
(with Prof. Ben Glazer, English)
 - Web site to catalog, view, and analyze Babylonian collection
(with Dr. Agnete Lassen, Associate Curator of Yale Babylonian collection)
 - Web site to curate public art on campus
(with Carol Snow, Deputy Chief Conservator of Yale University Art Galleries)
 - Assembling ancient wall paintings
(with Dr. Benedict Brown, Computer Science)
 - ...



CS 276: Applications in the Digital Humanities

First half of semester:

- Web server application development (e.g. Flask)
- Web application frameworks (e.g. JavaScript, AngularJS, ReactJS)
- SQL Database design
- Technical topics alternate with discussion of applications in digital humanities
- Midterm

Second half of semester

- Group work on a digital humanities project of your choice
- Guest lectures, interim and final project presentations

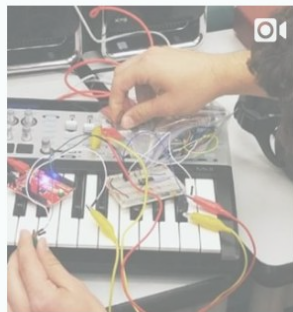
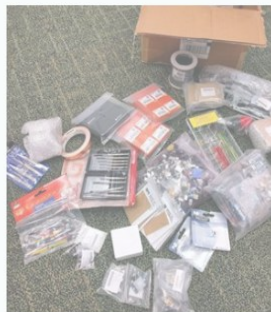
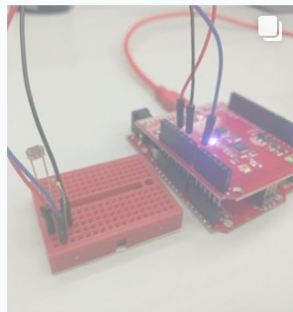
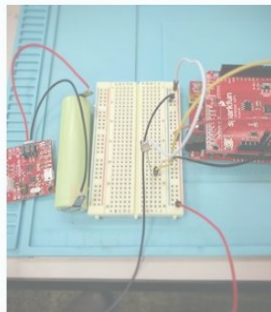
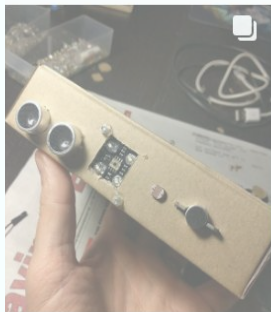
Open to anyone with *some* programming experience (CS50, CS 112, self-taught, etc.)

[illegible]

Scott Petersen

Lecturer, CS

OMI • CPSC 035, 134, 431, 432



Yale Open Music Initiative

exploring open source hardware and software at
the intersection of music, sound and technology.



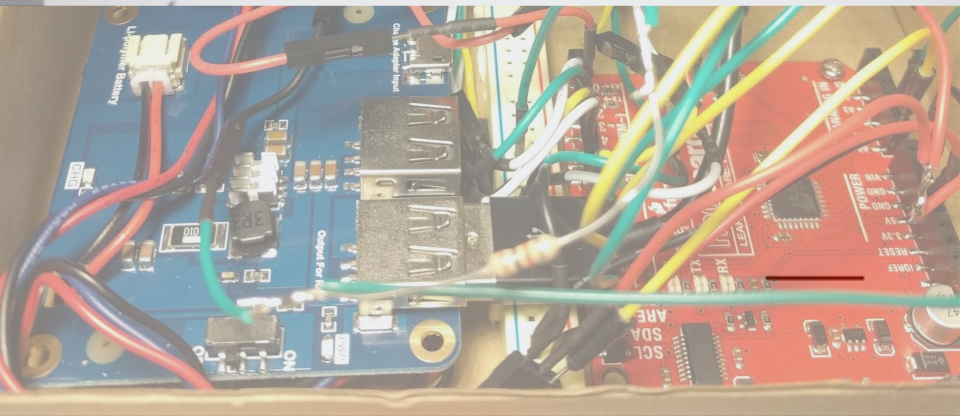
CPSC 134

Programming Music Applications





```
16 h = 90;  
17 a = 86;  
18 l = 8;  
19 w = Window("Row of trees", Rect(300, 300, 800, 400)).front;  
20 u = UserView(w, w.view.bounds).background_(Color.black)  
21   .clearOnRefresh_(false)  
22   .drawFunc_  
23   ({  
24     var dx, dy, nextP;  
25     if(c,
```



CPSC 035

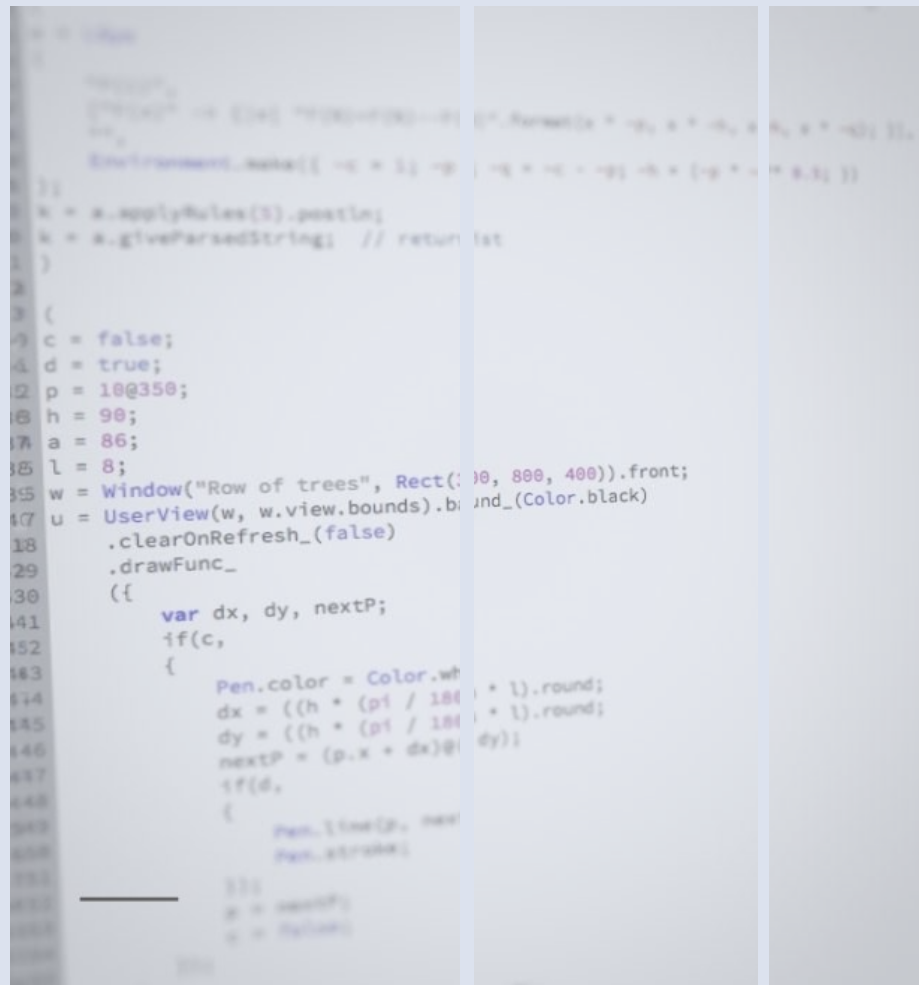
21st Century

Computer Music

Questions?

- CPAR Music Track
- Open Music Initiative
- Music & Tech Classes @ Yale
- AKW Music Studio Tour

scott.petersen@yale.edu



CS 200b / CS 201b

Stephen Slade

Biography. Hybrid: academia (theory) / industry (practice)

- Yale undergraduate. music major. (studied CS with Alan Perlis)
- Developed computer systems for presidential campaigns / White House.
- Yale graduate student, M.S., Ph.D., computer science (artificial intelligence)
- Yale CS Department: Asst Chairman, Asst Director of the Yale AI Project.
- NYU Stern School of Business, Department of Information Systems, Assistant Professor.
- Wrote three books (T/Scheme, LISP, decision making) - dedicated to Yale alumnae.
- Developed investment and risk technology systems for Wall Street. Sell side (Morgan Stanley), Buy side (INVESCO, Bank of America, Commonfund)

The intersection of the two courses

Both courses cover:

- Recursion (what happens when you Google “recursion”?)
- UNIX (using the zoo)
- Computer Architecture - machine language
- Programming assignments every week or so
- Logical problems - thinking outside the box

Logical Problem Example

What is the next number in the follow sequence?

1969, 1973, 1977, 1981, 1985, 1991, 1993, 1999, ...

CS 201 - focus is on theory

- Programming language: Racket (a dialect of Scheme / LISP)
- Turing Machines
- Boolean logic
- Digital gates and circuits
- Machine language for a fictional machine architecture (TC 201)
- Formal languages, regular expressions, deterministic finite automata
- Computation complexity and analysis of algorithms
- http://zoo.cs.yale.edu/classes/cs201/Fall_2017/syllabus.html
- Satisfies *de jure* requirements for major, CS 223, etc.

CS 200 - focus is on practice

- Programming language: Python
- Object-oriented programming (like Java and C++)
- Software engineering: how to write, debug, and maintain code.
- Databases, building a database system and using SQL
- Machine language for Python Virtual Machine
- Cryptography and information security
- Machine learning, aka, big data or data science
 - e.g., driverless cars or cryptanalysis
- <http://zoo.cs.yale.edu/classes/cs200/syllabus.html> (last spring)
- Satisfies *de facto* requirements for CS 223, with instructor's permission