

This is CS 50.

This is CS 50.



Harvard College's Introduction to Computer Science I

# COMPUTER SCIENCE 50

---

**WEEK 0**

**DAVID J. MALAN '99**

malan@post.harvard.edu

# Divide and Conquer





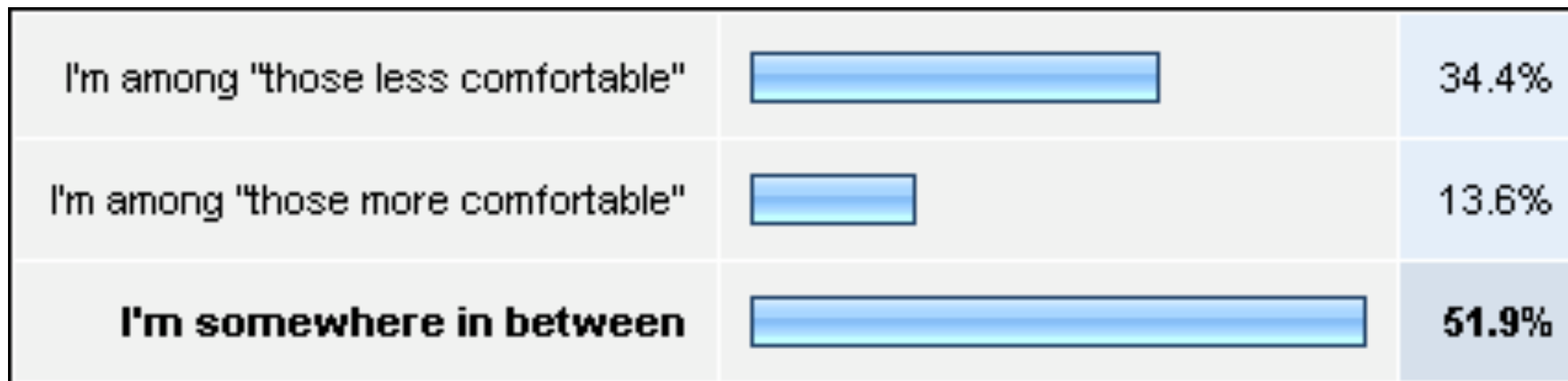
# Lolcats



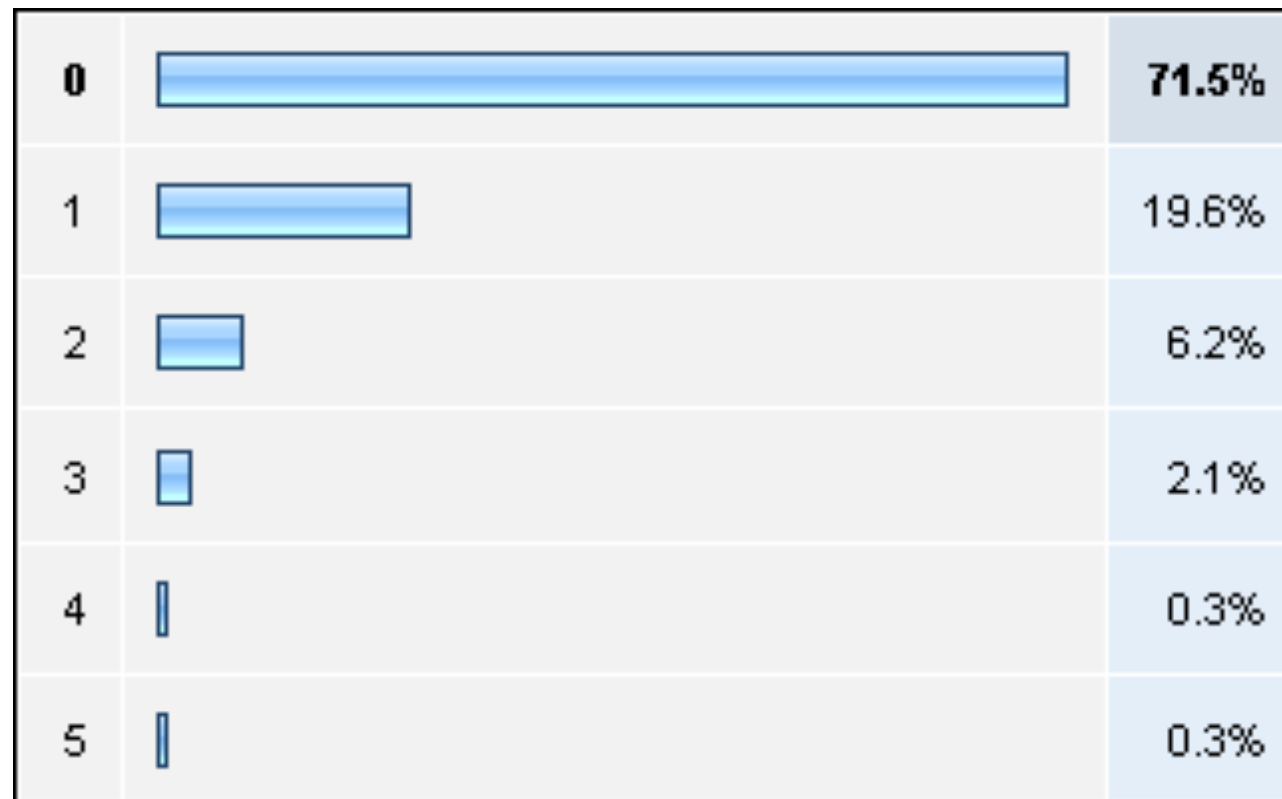
# Roll Call

1. Stand up.
2. Think to yourself: "I am #1."
3. Pair off with someone standing, add your numbers together, and adopt the sum as your new number.
4. One of you should sit down, the other should go back to step 3.

# “Does everyone know more than me?”



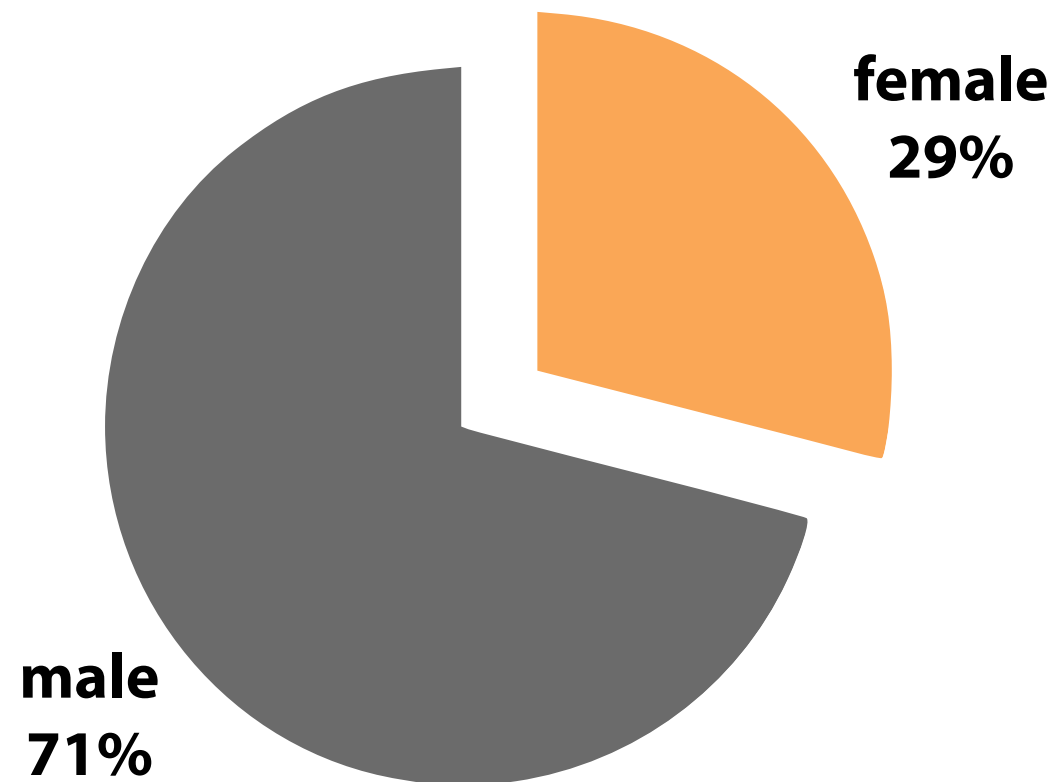
# Prior Coursework in CS



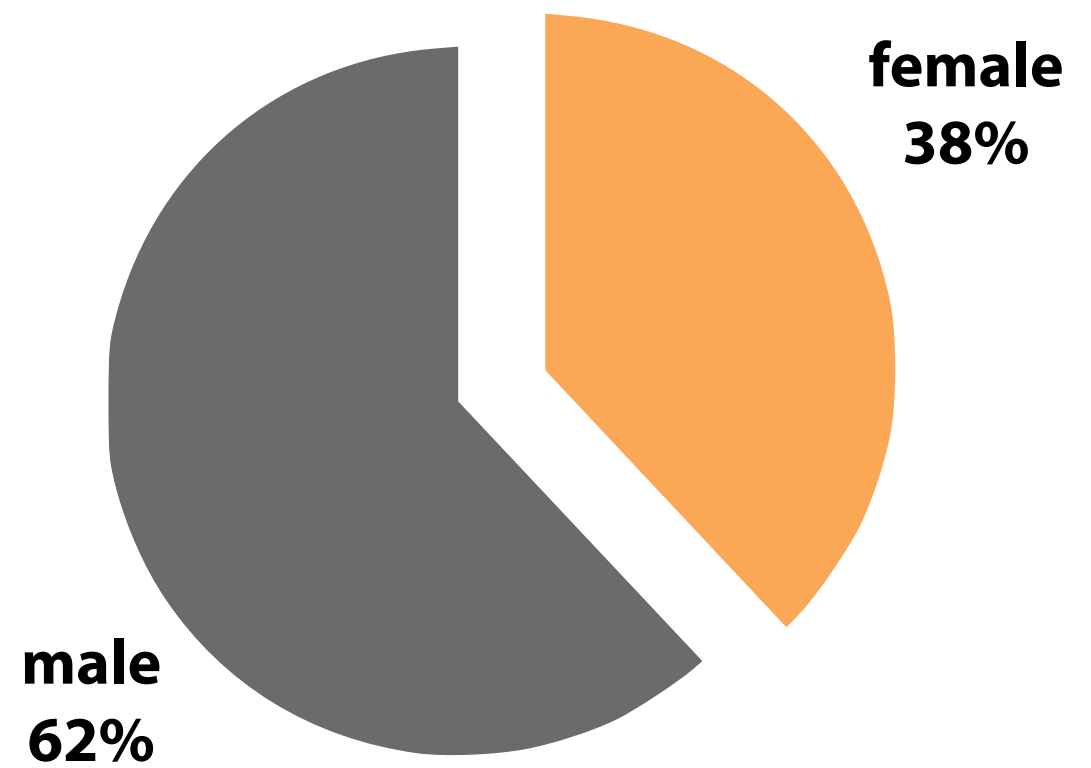


# Gender Ratio

**Fall 2007**



**Fall 2008**





*20th Anniversary Edition*

★ 92 weight  
★ 20.

5000 sheets  
letter size - 8 1/2 x 11-  
216 x 279 mm  
acid free  
Item# 135848

n00b



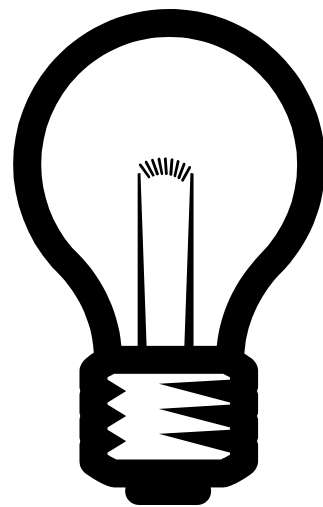
133+







# Binary



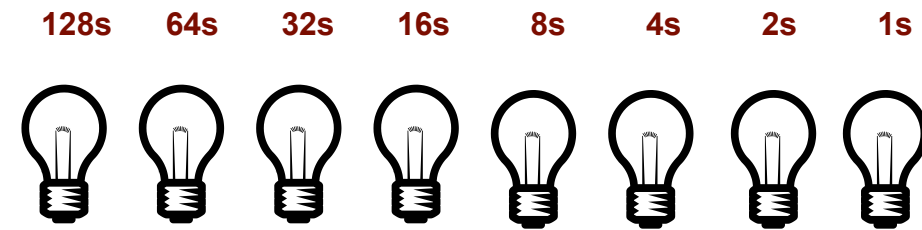
# Binary



# Binary

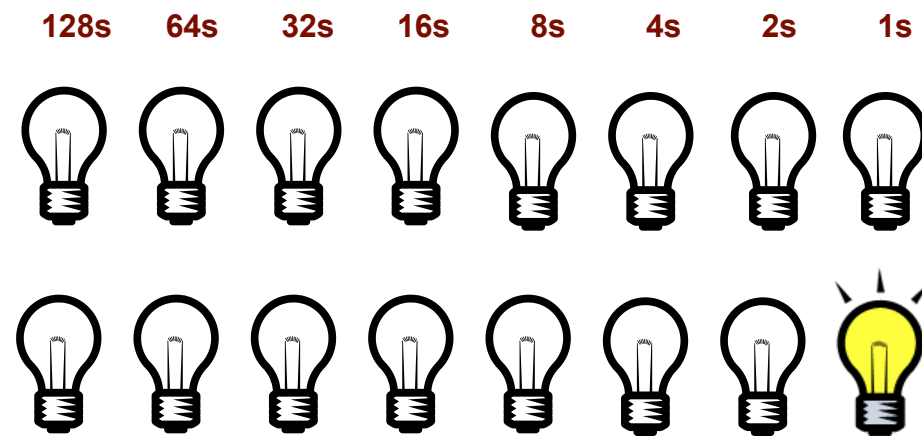
128s 64s 32s 16s 8s 4s 2s 1s

# Binary

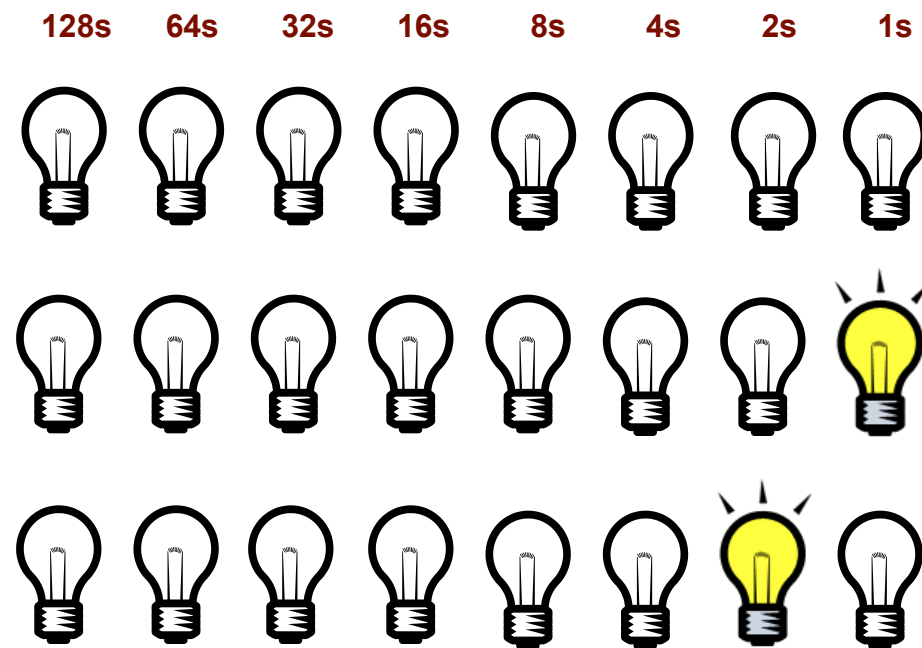




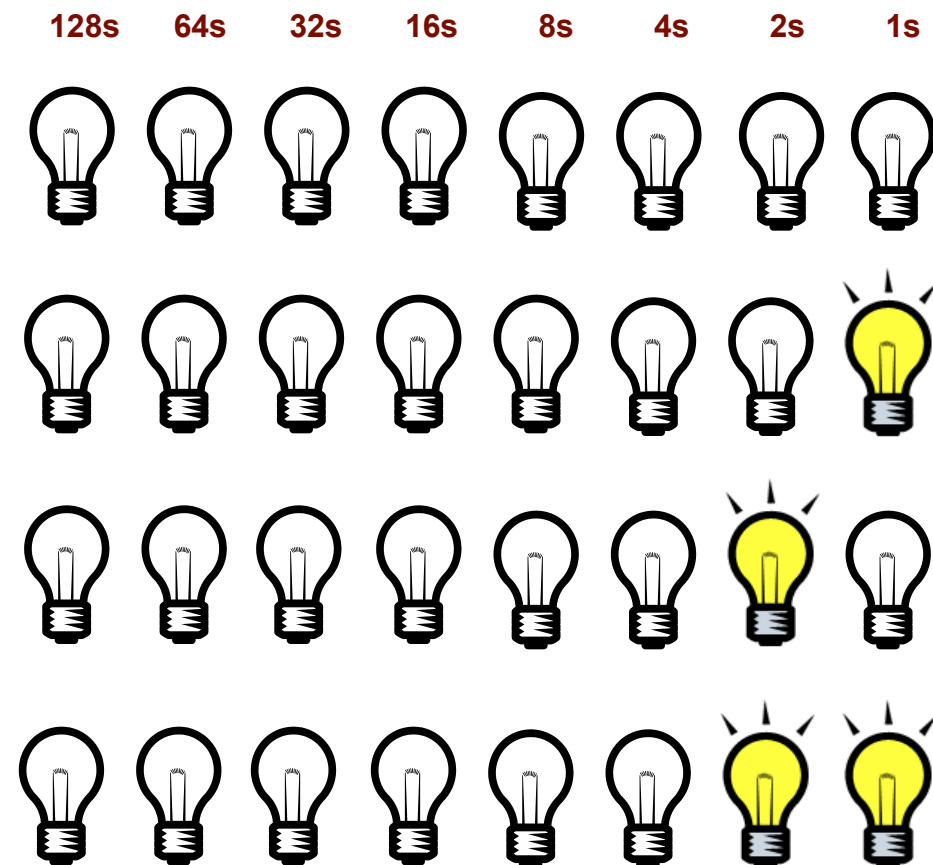
# Binary



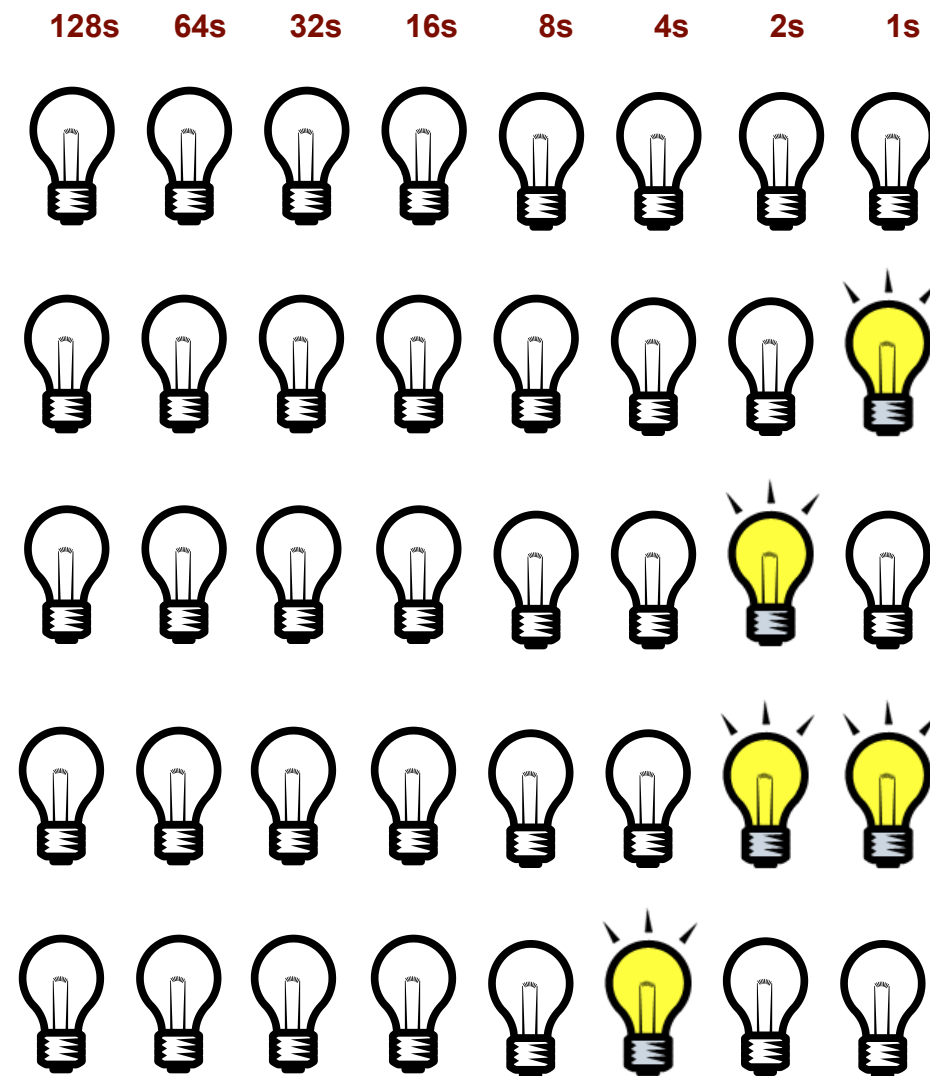
# Binary



# Binary



# Binary



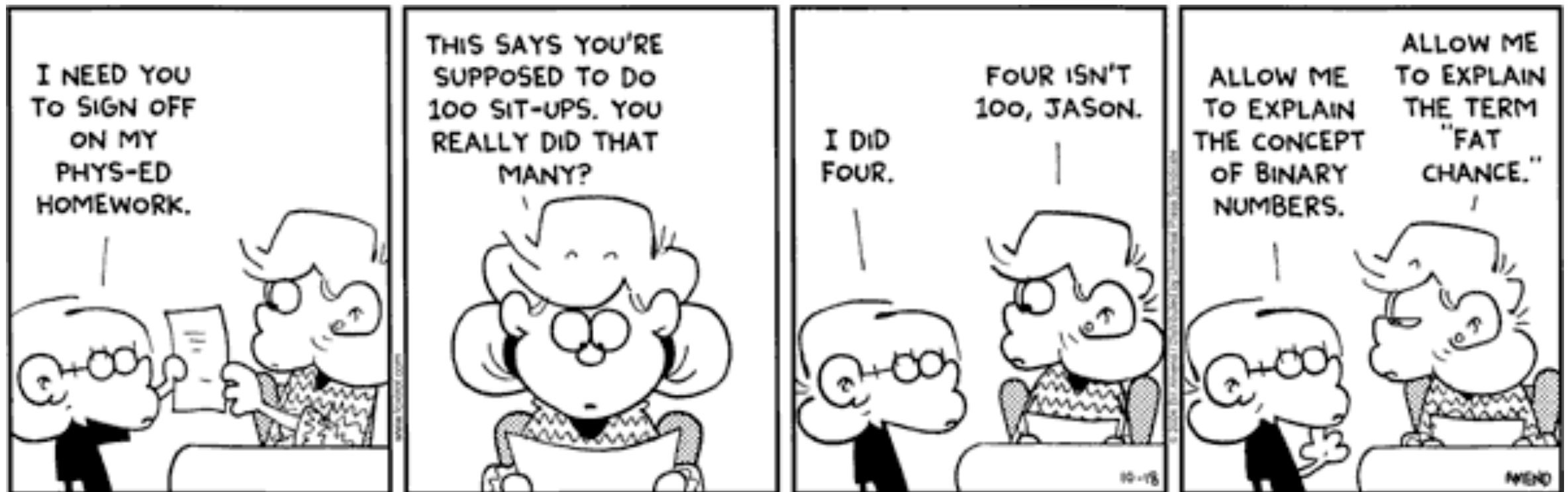


# ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b>@</b>	96	60	140	&#96;	<b>`</b>
1	1	001	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2	002	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3	003	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4	004	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5	005	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6	006	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7	007	<b>BEL</b> (bell)	39	27	047	&#39;	<b>'</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8	010	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9	011	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A	012	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B	013	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C	014	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D	015	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E	016	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F	017	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10	020	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11	021	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12	022	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13	023	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14	024	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15	025	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16	026	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17	027	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18	030	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19	031	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A	032	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B	033	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>:</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C	034	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D	035	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E	036	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F	037	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

Source: [www.asciitable.com](http://www.asciitable.com)

# Geek Humor



# Lectures

## Week 0

Introduction. Bits. Binary. ASCII.  
Programming. Algorithms. Scratch.  
Statements. Boolean expressions. Conditions.  
Loops. Variables. Threads. Events.





# Lectures

## Week 1

Functions, continued. Global variables. Parameters. Return Values. Stack. Frames. Scope. Arrays. Strings. Command-line arguments. Cryptography.

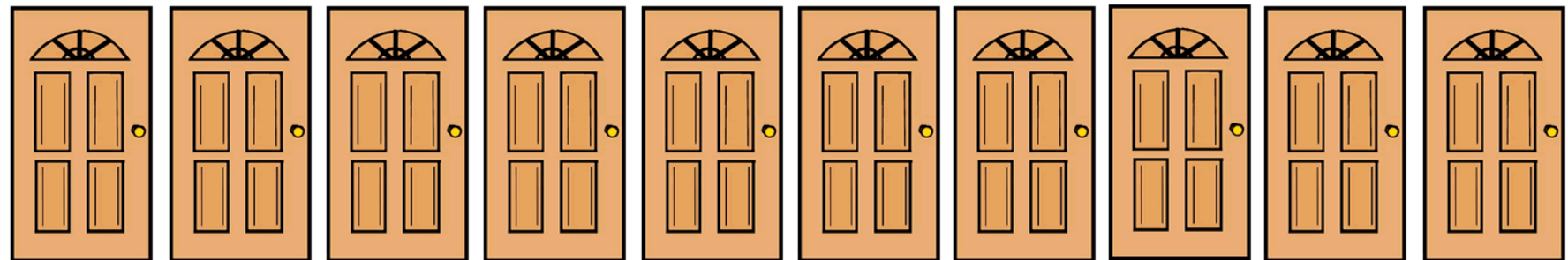




# Lectures

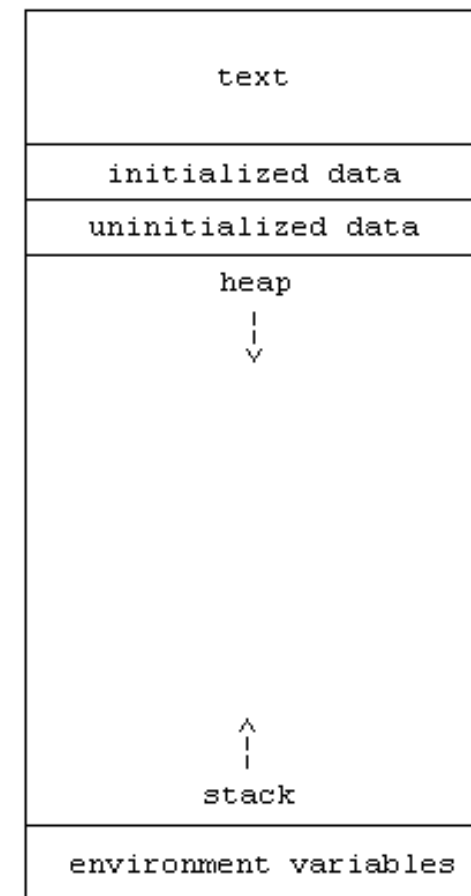
## Week 3

Linear search. Binary search. Asymptotic notation. Recursion. Pseudorandomness. Bubble sort. Selection sort. Insertion sort. Merge sort. Debugging.



# Week 4

Structures. Dynamic memory allocation. Stack and heap. Pointers. Debugging, continued.



# Lectures

## Week 5

File I/O. Forensics. Linked lists. Stacks.  
Queues.

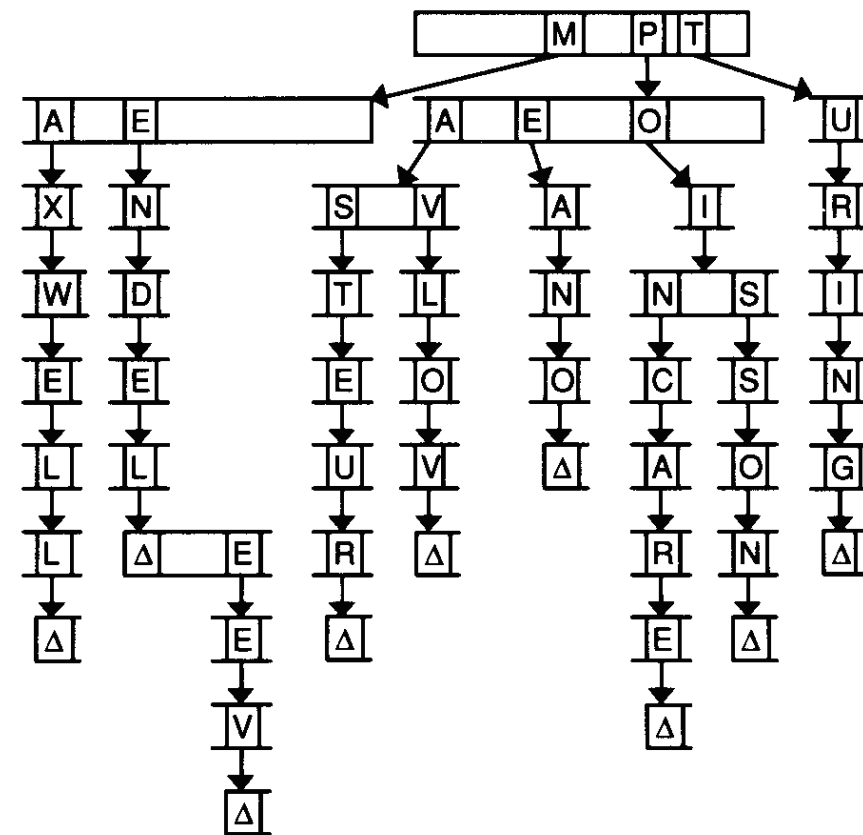




# Lectures

## Week 7

Valgrind. Bitwise operators. Hash tables.  
Trees. Binary search trees. Tries. Heaps.  
Heapsort. Huffman coding.

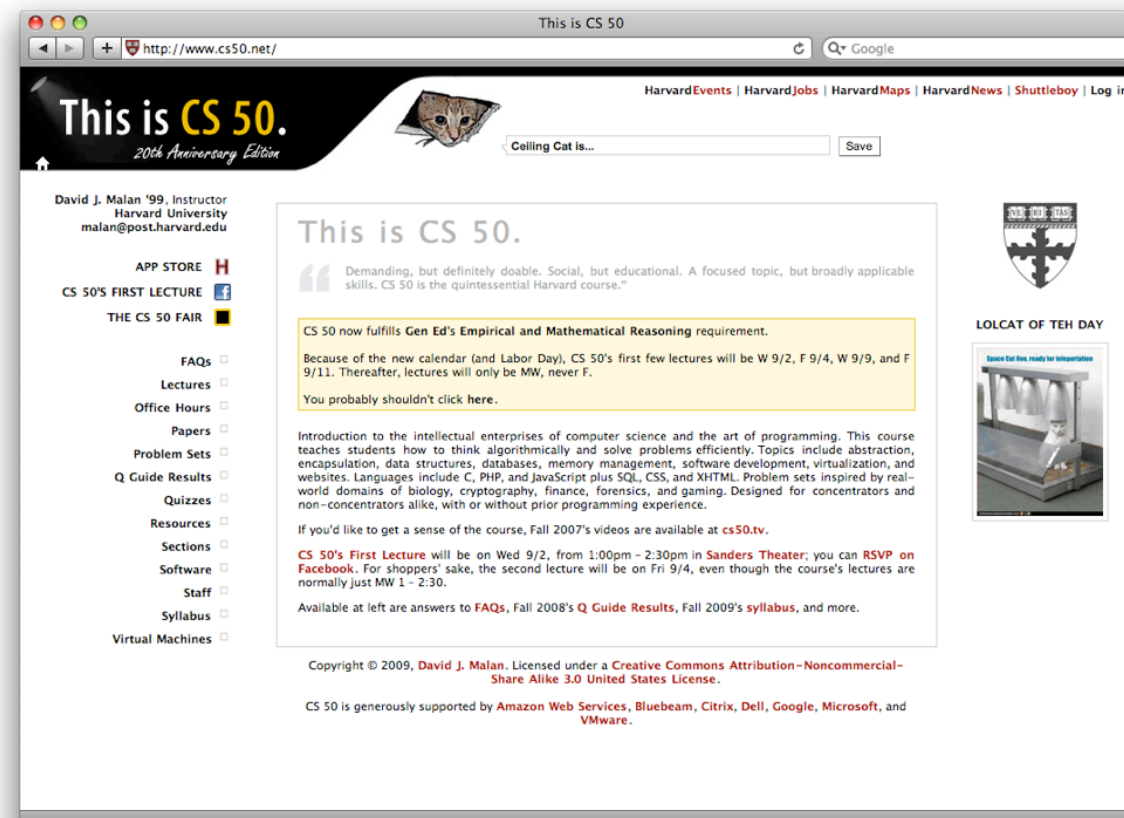




# Lectures

## Week 8

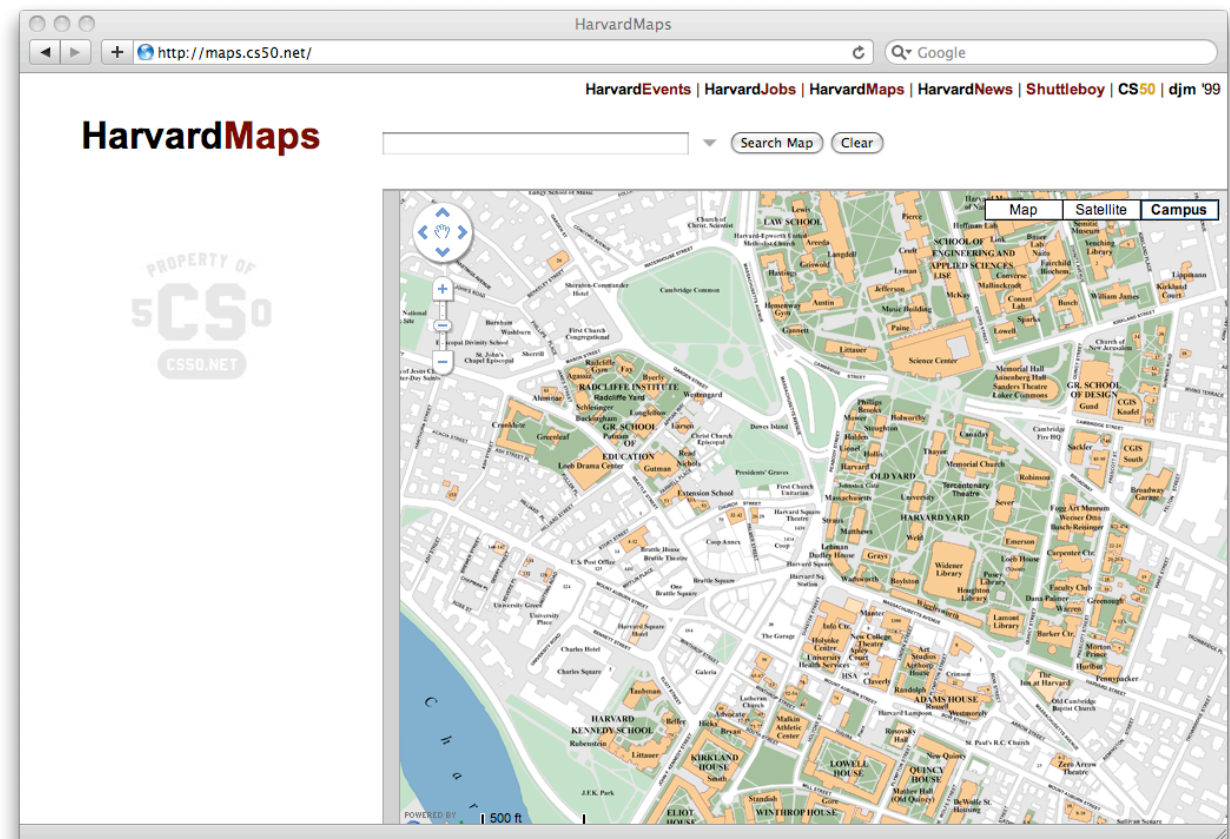
TCP/IP. HTTP. XHTML. PHP. SQL.



# Lectures

## Week 9

DOM. CSS. Inheritance. JavaScript. Events, continued. OOP. Ajax.



# Lectures

## Week 10

Preprocessing. Compiling. Assembling.  
Linking. CPUs.





# Expectations

- ▶ Attend all lectures and sections.
- ▶ Complete nine problem sets.
- ▶ Take two quizzes.
- ▶ Produce a final project.
- ▶ (No final exam.)



# Grades

- ▶ Problem Sets (best 8 out of 9): **60%**
- ▶ Quizzes: **30%**
- ▶ Final Project: **10%**

You may take the course pass/fail.

# Website

<http://www.cs50.net/>

- ▶ Bulletin Board
- ▶ Grades
- ▶ Handouts
- ▶ Resources
- ▶ Software
- ▶ Videos
- ▶ Virtual Terminal Room
- ▶ ...

# Books

**None are required!**

- ▶ Absolute Beginner's Guide to C
- ▶ Programming in C
- ▶ How Computers Work
- ▶ Hacker's Delight

# Lectures

- ▶ MW 1 – 2:30pm.
- ▶ Plus this F and next F.





# Sections

**Online sectioning begins Wed 9/9!**

- ▶ For “those less comfortable.”
- ▶ For “those more comfortable.”
- ▶ For those somewhere in between.

# Staff

- ▶ Teaching Fellows
- ▶ Course Assistants
- ▶ Sysadmins
- ▶ AV, Producers, Videographers
- ▶ Scribes
- ▶ me



# Office Hours

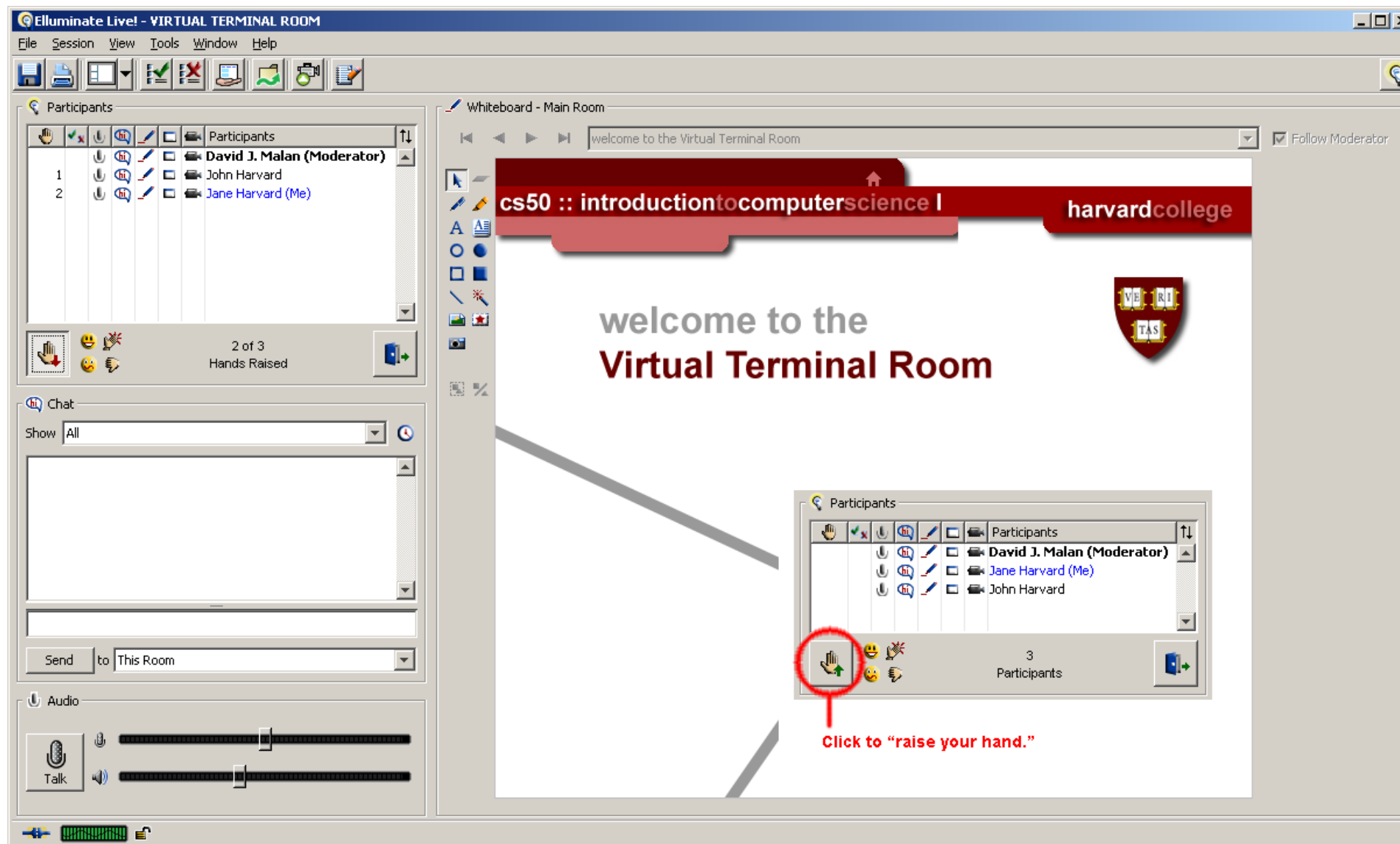
Start next week!



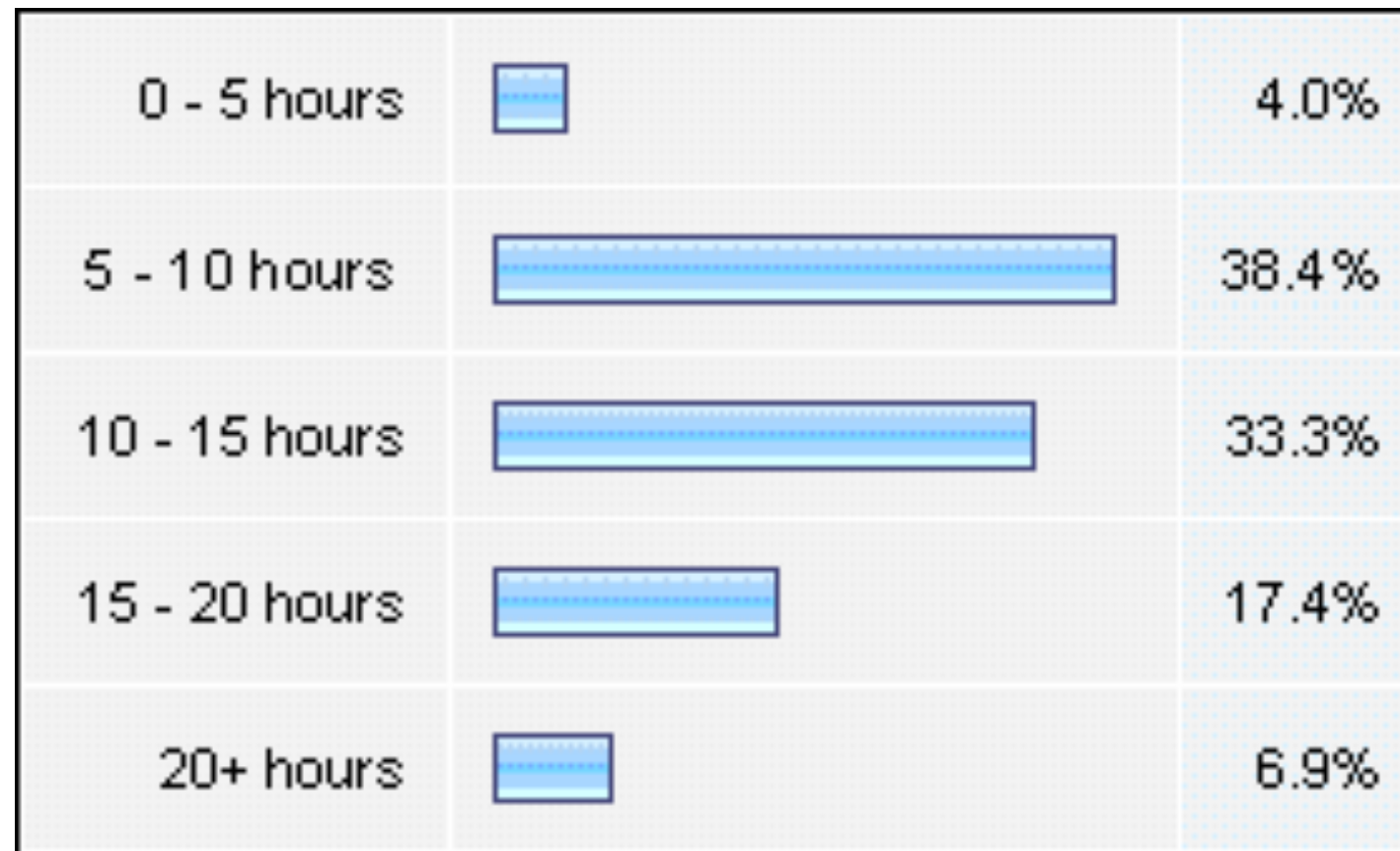


# Virtual Office Hours

## Start in two weeks!



# Workload

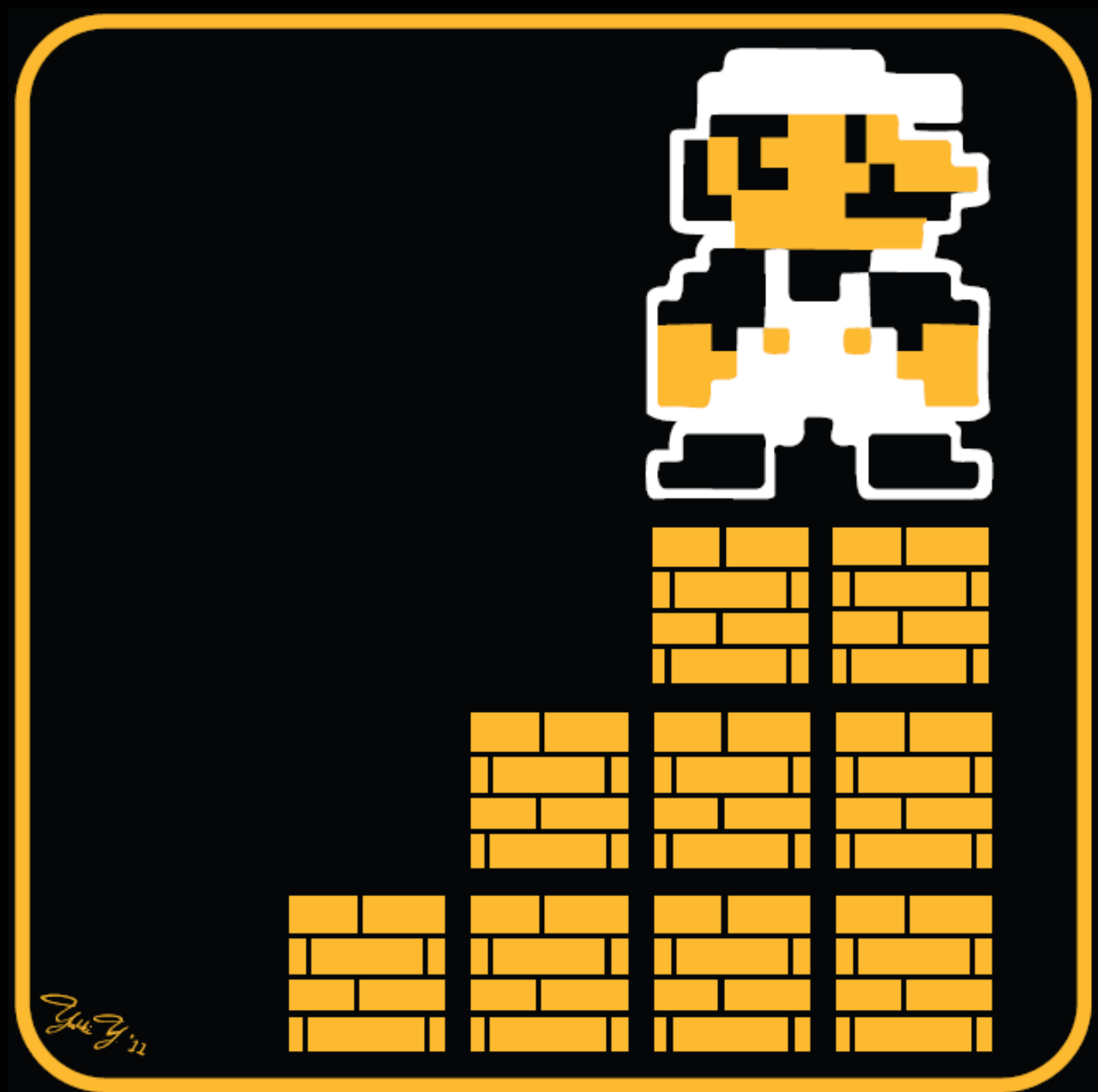




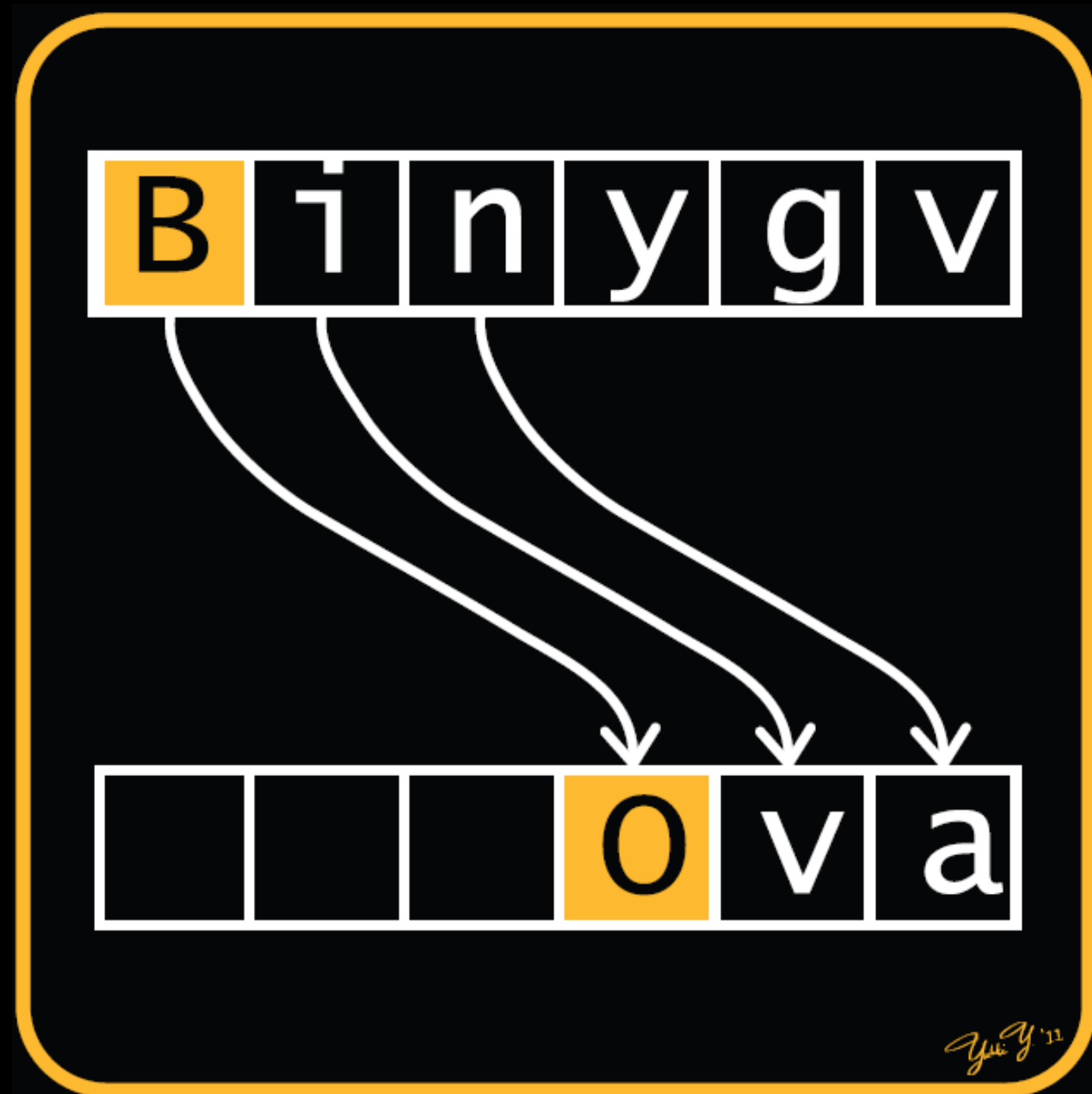
*Yui Y. '11*

**0: Scratch**





1:C



2: Crypto



### 3: Game of Fifteen



# 数独

8			4		6			7
						4		
	1					6	5	
5		9		3		7	8	
				7				
	4	8		2		1		3
	5	2					9	
		1						
3			9		2			5

2022.9.12

## 4: Sudoku



## 5: Forensics





Teh Computer Sci-  
ence 50 learnz you  
about intertubes in  
teh cloudz. David J.  
Malan will pwn u  
ftw so watch out!

*ayy*

## 6: Misspellings

**CS50**

↑ **120**



**7: C\$50 Finance**



## 8: Mashup



THE

CS 50

FAIR

This is CS 50.

# Algorithms

```
1  let socks_on_feet = 0
2  while socks_on_feet != 2
3      open sock drawer
4      look for sock
5      if you find a sock then
6          put on sock
7          socks_on_feet++
8          look for matching sock
9          if you find a matching sock then
10             put on matching sock
11             socks_on_feet++
12             close sock drawer
13         else
14             remove first sock from foot
15             socks_on_feet--
16     else
17         do laundry and replenish sock drawer
```



# 0 hai, C!

hai.c

```
#include <stdio.h>

int
main(int argc, char *argv[])
{
    printf("0 hai, world!\n");
}
```

# 0 Hai, C!

```
#include <stdio.h>
```

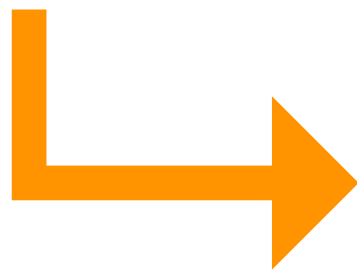
```
int
```

```
main(int argc, char *argv[])
```

```
{
```

```
    printf("0 hai, world!\n");
```

```
}
```



```
10000011 00000001 00010001 00000000 00111101 11111100 01110100 00111101
00000000 01000000 00000000 00000000 00000000 00000000 00000000 00000000
10010000 00000000 00000000 00000000 01010000 00000000 00000111 00110000
00001011 00000001 00001011 00000011 00001010 00000000 00000000 00000000
00000000 00100000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00100000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
01110000 00010000 00000000 00100000 00000001 00000000 00000000 00000000
00000000 00000000 00000000 00100000 00000001 00000000 00000000 00000000
00000000 00000000 00000000 01000000 00000001 00000000 00000000 00000000
00000000 00100000 00000000 01000000 00000001 00000000 00000000 00000000
11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
10010000 10000000 00000000 01000000 00000001 00000000 00000000 00000000
00101110 01100100 01111001 01101110 01100001 01101101 01101001 01100011
10110000 00000100 00000000 00100000 00000001 00000000 00000000 00000000
10110000 00000100 00000000 00100000 00000001 00000000 00000000 00000000
10100000 00000001 00000000 00000000 00000000 00000000 00000000 00000000
10110000 00000100 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00100000 00000000 00000000
```

[...]

# 0 hai, Scratch!

Hai1.sb





# Statements

say O hai, world!

wait 1 secs

play sound meow ▼

...

# Statements

Hai{2,3}.sb

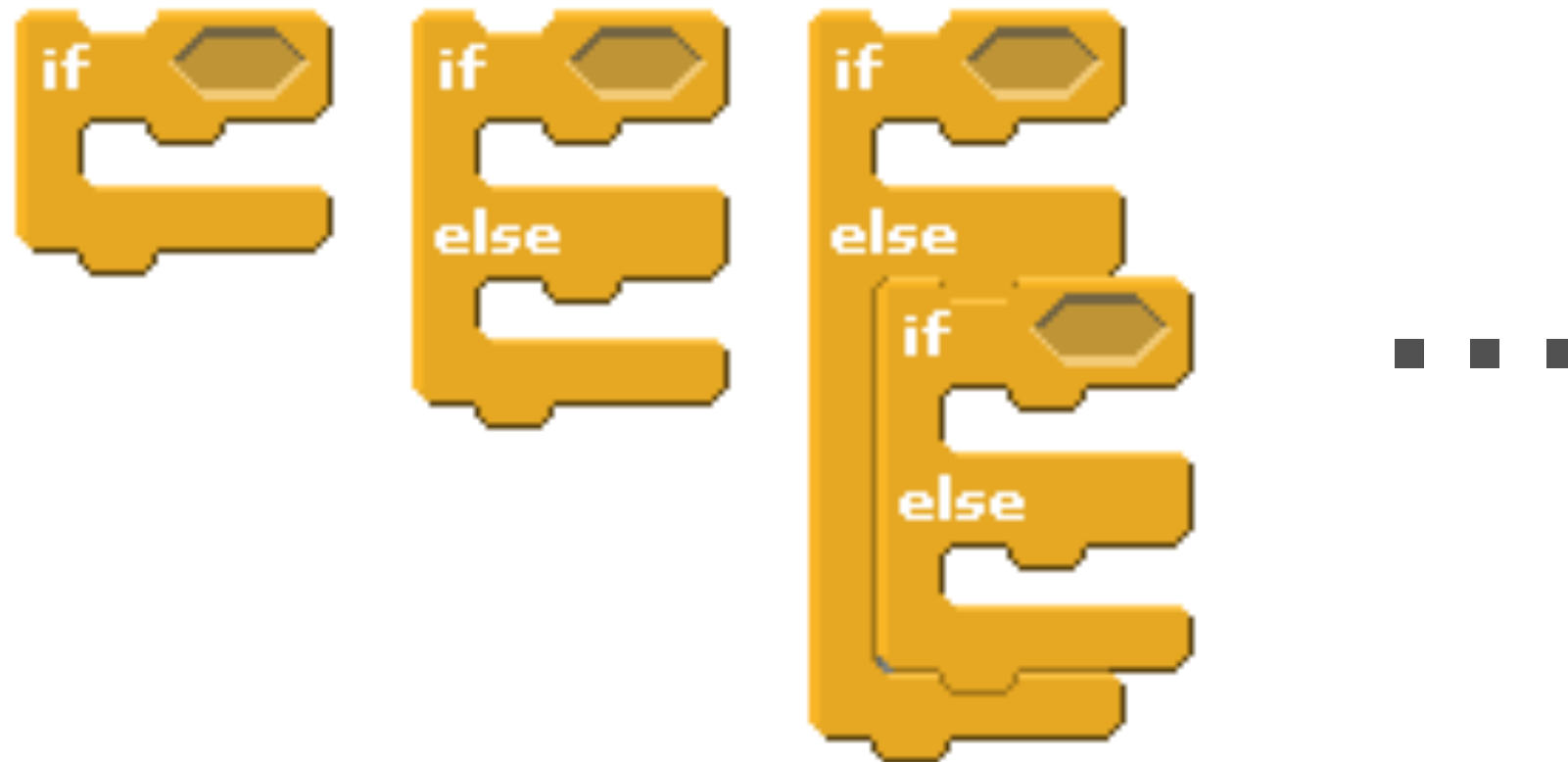


# Boolean Expressions



■ ■ ■

# Conditions





# Conditions

Hai{4,5}.sb



# Loops



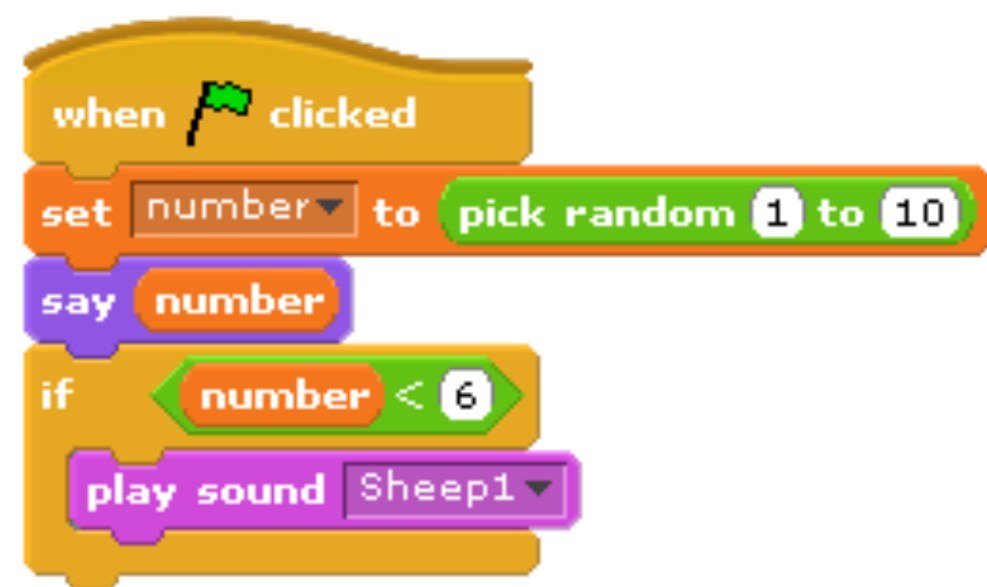
# Loops

Hai{6,7,8}.sb



# Variables

Count{1,2}.sb





# Arrays

add thing to inventory ▼

delete 1 ▼ of inventory ▼

insert thing at 1 ▼ of inventory ▼

replace item 1 ▼ of inventory ▼ with thing

item 1 ▼ of inventory ▼

length of inventory ▼

# Arrays

FruitcraftRPG.sb



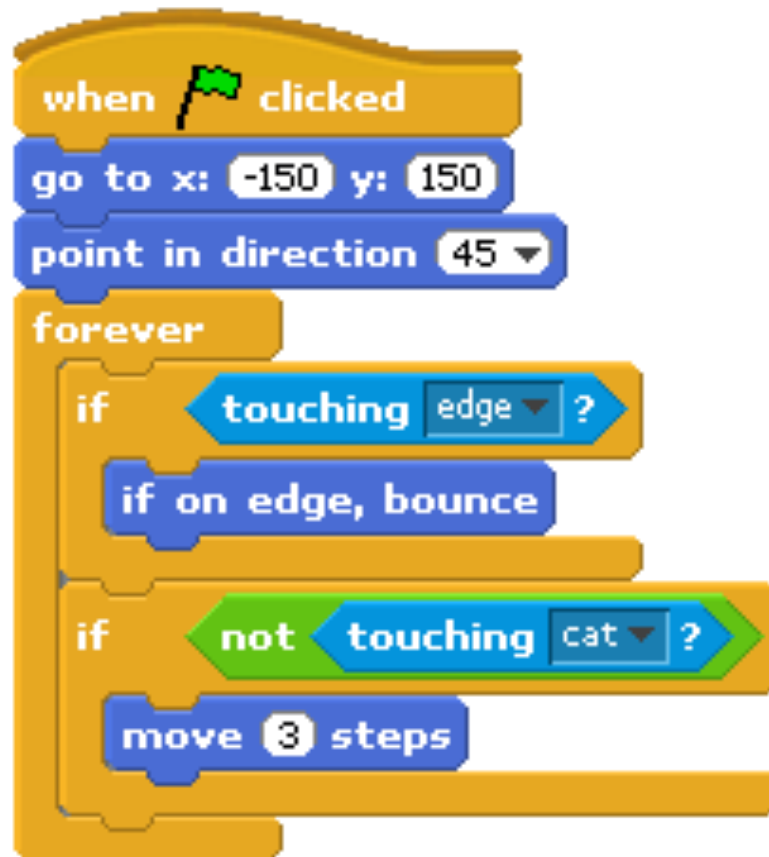
# Threads

## Move1.sb



# Threads

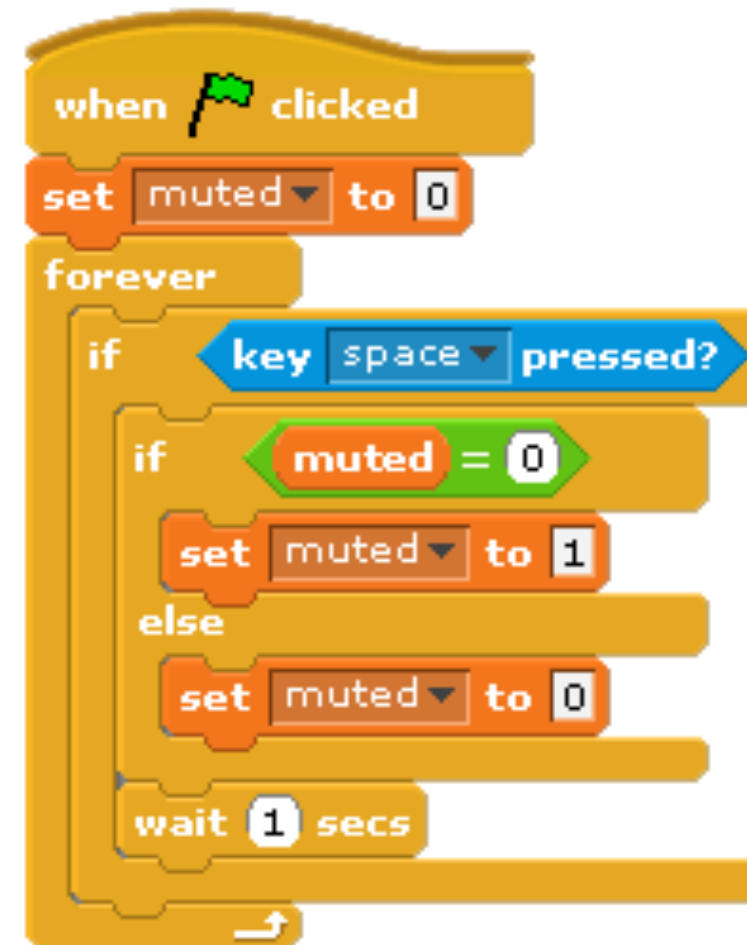
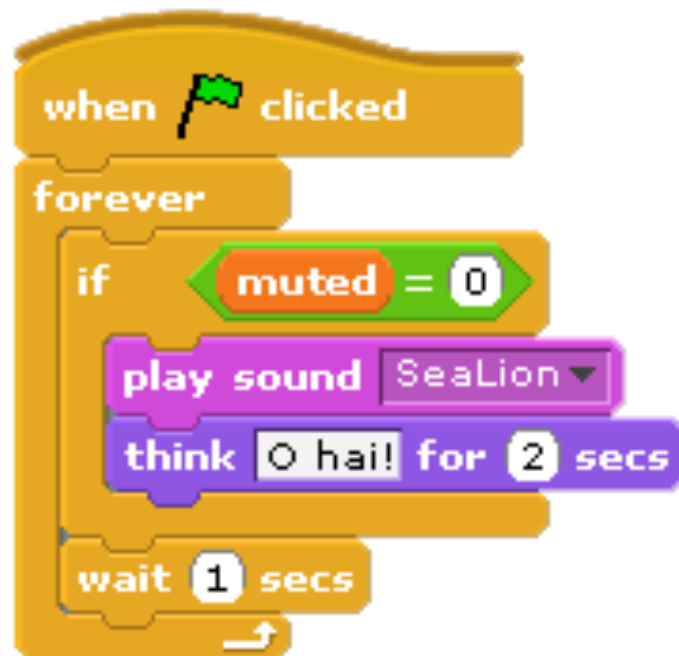
## Move2.sb





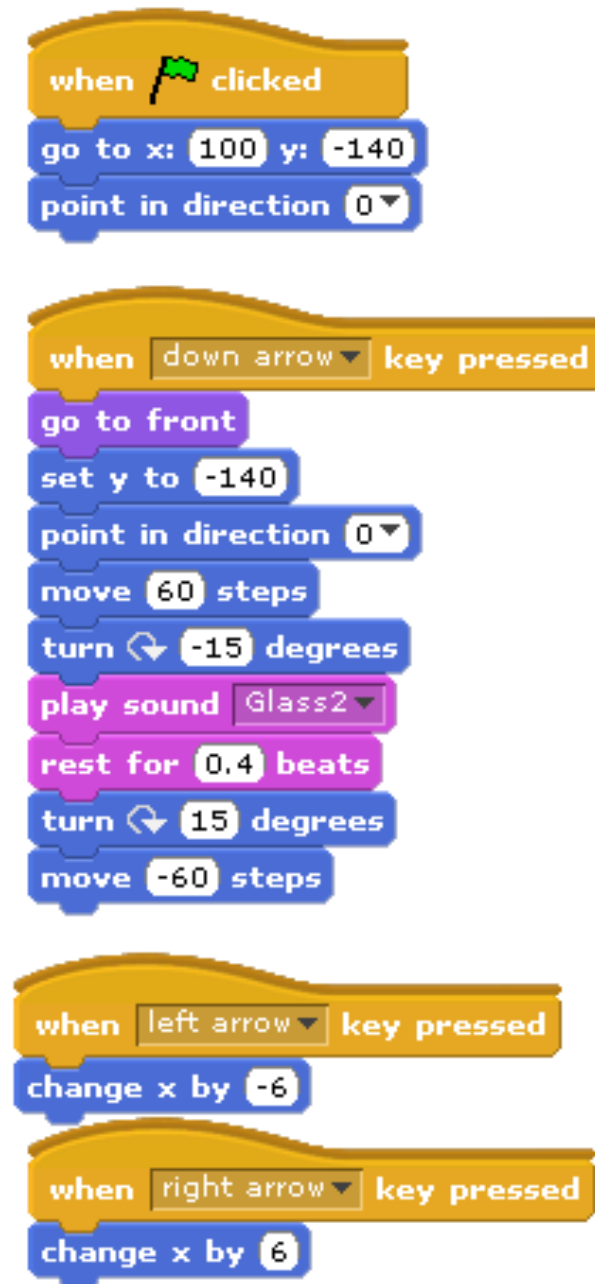
# Threads

Hai10.sb



# Threads

David.sb



# Events

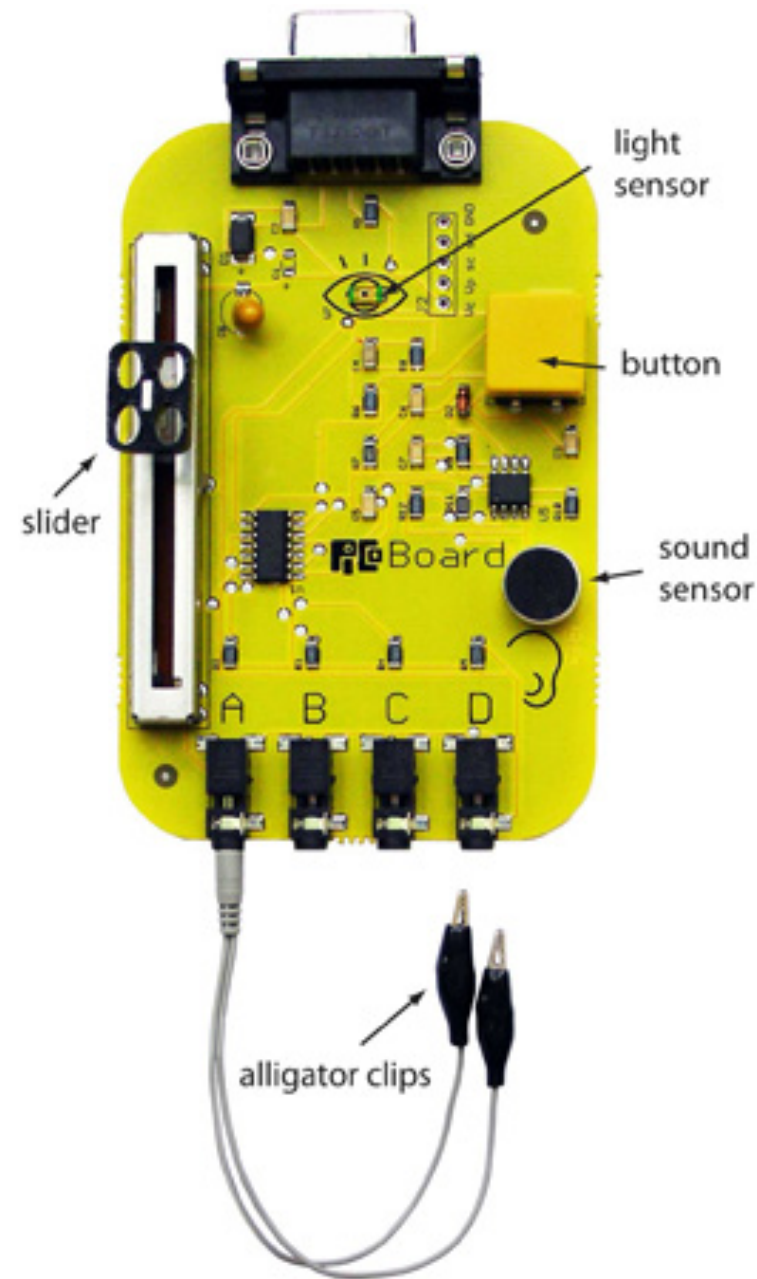
Marco.sb





# Sensors

singer.sb, Masquerade.sb, davidwu.sb





# Oscartime

Oscartime.sb

