Resources and opportunities…

http://biblio.roboteducation.org/

IPRE's extensive robot-in-education bibliography

AAAII 2008 Robot exhibition

email Paul Oh this week to join in!

AAAII 2008 AI Education Workshop

AI in a breadth-first CS 1

Zachary Dodds

3/27/08  -  AAAI Spring 2008 Stanford
Four undergraduate years
   ~ 190 students each

Every student must pass CS 1

Wanted from CS 1:
   more (women) CS majors
   more enthusiasm
   more CS
CS 1 for scientists

150 lecture minutes, lab

CS breadth

<table>
<thead>
<tr>
<th>Weeks</th>
<th>Paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>functional</td>
</tr>
<tr>
<td>4-6</td>
<td>machine-level</td>
</tr>
<tr>
<td>7-9</td>
<td>imperative</td>
</tr>
<tr>
<td>10-12</td>
<td>objects+classes</td>
</tr>
<tr>
<td>13-15</td>
<td>theory/projects</td>
</tr>
</tbody>
</table>

breadth-first (and -last)
CS 1 for scientists

2-3 lectures per week, 1 lab
two experience-based sections

CS breadth  →  Al breadth

<table>
<thead>
<tr>
<th>Weeks</th>
<th>Paradigm</th>
<th>AI-themed labs and assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>functional</td>
<td><code>turing()</code>, <code>Caesar decipher</code></td>
</tr>
<tr>
<td>4-6</td>
<td>machine-level</td>
<td><code>audio classification</code></td>
</tr>
<tr>
<td>7-9</td>
<td>imperative</td>
<td><code>Markov text generation</code></td>
</tr>
<tr>
<td>10-12</td>
<td>objects+classes</td>
<td><code>Connect Four</code></td>
</tr>
<tr>
<td>13-15</td>
<td>theory/projects</td>
<td><code>robotic navigation</code></td>
</tr>
</tbody>
</table>

Small but recurring theme: 6 of the ~42 hw problems
Functions first  

CS

- conditionals & variables
- functions & recursion

AI

- language: meaning
- vs. mechanics
- computer conversationalist

programs can lie...

def turing():
    choice = raw_input("Rock, paper, or scissors? ")
    print "You chose scissors."
    print "I chose rock"
    print "You lose!"

but can they act contrary to their programming?
With acknowledgments to Tim…
Our own *functioning*

*machino*-morphizing
### Turing and Loebner

**Judge 4 John Sundman**

<table>
<thead>
<tr>
<th>Round 2</th>
<th>Carpenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milardo</td>
<td>Score: 92, Rank: 2</td>
</tr>
<tr>
<td></td>
<td>Score: 8, Rank: 7, Program</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Round 3</th>
<th>Wallace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armour</td>
<td>Score: 90, Rank: 3</td>
</tr>
<tr>
<td></td>
<td>Score: 10, Rank: 6, Program</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Round 6</th>
<th>Watkins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Espinales</td>
<td>Score: 98, Rank: 1</td>
</tr>
<tr>
<td></td>
<td>Score: 2, Rank: 8, Program</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Round 7</th>
<th>Veselov</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salamensky</td>
<td>Score: 70, Rank: 4</td>
</tr>
<tr>
<td></td>
<td>Score: 30, Rank: 5, Program</td>
</tr>
</tbody>
</table>

Scores for confederates and programs.
The philosophy...

AI provides problems with compelling contexts and motivations…

already provided

success with hw problem

extra credit

and beyond!

push

grades expectations

pull

self-perception challenge

big ideas
elif choice == 'case':
    print 'Case is one of my favorite dorms. It\'s design is such
    that the only entrance is the only exit. This, in conjunction with its thick walls, has given it the
    nickname "The Dungeon," because no one can hear the screams.'
    print

choice = raw_input('Would you like to visit Nate Jones?
print

if choice[0] == 'y':
    print 'Good choice. He is the head of a metal band named "The
    Black Ravens of Immortal Death." He might get pissed if you pass him by.'
    print

elif choice[0] == 'n':
    print 'Well, fine then. Don\'t visit him.'
    print
elif choice[1:] == 'ase':
    print 'Case is one of my favorite dorms. It\'s design is such that\nthe only entrance is the only exit. This, in\nconjunction with its thick walls, has\ngiven it the nickname "The Dungeon,"nbecause no one can hear the screams.'

choice = raw_input('Would you like to visit Nate Jones?')
print

if choice[0] == 'y':
    print 'Good choice. He is the head of a metal band named\n"The Black Ravens of Immortal Death." He might get pissed if you\npass him by.'

print

elif choice[0] == 'n':
    print 'Well, fine then. Don\'t visit him.'

print

task-driven discoveries…
and beyond!

```python
elif choice == 'case':
    print 'Case is one of my favorite dorms. It\'s design is such that the only entrance is the only exit. This, in conjunction with its thick walls, has given it the nickname "The Dungeon," because no one can hear the screams.'

choice = raw_input('Would you like to visit Nate Jones?')

if choice[0] == 'y':
    print 'Good choice. He is the head of a metal band named "The Black Ravens of Immortal Death." He might get pissed if you pass him by.'

elif choice[0] == 'n':
    print 'Well, fine then. Don\'t visit him.'
```

I learn just as much!
Other work

weeks 1-3

recursive thinking: **map, reduce**...

sorting, selection

turtle graphics

random walks

"lights out"

Pig Latin translator

dot products, integration

scrabble scoring
 Functional Finale

Caesar deciphering

```python
>>> encipher('Caesar cipher? I prefer Caesar salad.', 25)
'Bzdrzq bhogdq? H oqdedq Bzdrzq rzkzc.'

>>> decipher('Hu lkbjhapvu pz doha ylthpuz hmaly dl mvynla '\
            'lclyfaopun dl ohcl slhyulk."
'An education is what remains after we forget everything we have learned.'

>>> decipher('gv vw dtwvg')
```
Functional Finale  Caesar deciphering

>>> decipher('Bzdrzq bhogdq? H oqdedq Bzdrzq rzkzc.')
'Caesar cipher? I prefer Caesar salad.'

>>> decipher('Hu lkbjhapvu pz doha ylthpuz hmaly dl mvynla \lcllyfaopun dl ohcl slhyulk.')
'An education is what remains after we forget everything we have learned.'

>>> decipher('gv vw dtwvg')

correct all encipherings

gv vw dtwvg
hw wx euxwh
ix xy fvyxi
jy yz gwzyj
kz za hxazk
la ab iybal
mb bc jzcbm
nc cd kadcn

od de lbedo
pe ef mcfep
qf fg ndgfq
rg gh oehgr
sh hi pfihs
ti ij qgjit
uj jk rhkju
vk kl silkv
wl lm tjmlw
xm mn uknmx
yn no vlony
zo op wmpoz
ap pq xnqpa
bq qr yorqb
cr rs zpsrc
ds st aqtsd
et tu brute
fu uv csvuf
Quantifying *Englishness*

approach is open-ended
we provide letter-frequencies

the best solutions don't use them!

**scrabble score** → **you're joking Mr. Feynman**

**iyebo tyusxq Wb. Poixwkx!**

**letter frequencies** → **sioly diecha Gl. Zyshguh**
Quantifying **Englishness**

approach is open-ended
we provide letter-frequencies
the best solutions don't use them!

You're joking Mr. Feynman

Iyebo tyusxq Wb. Poixwkx!

Sioly diecha Gl. Zyshguh

**Challenge**  Come up with a sentence for which native speakers would easily agree, but your system will fail.

I don't think qzxqxxz or zzzqzzqzx are words at all
Not everyone succeeds

tradition of highlighting creativity in all of its forms...

def encipher( phrase, shift ):
    code for shifting forward through the alphabet

def decipher( phrase ):
    """ This works sometimes! """
Not everyone succeeds

--- tradition of highlighting creativity in all of its forms... ---

def encipher( phrase, shift ):

code for shifting forward through the alphabet

def decipher( phrase ):

    """ This works sometimes! ""

    return encipher( phrase, 3 )

but it's OK...
Representation  
weeks 4-6

- base-N representations
- images and compression
- logic gates: AND OR NOT
- circuit composition
- assembly programming

CS  
(non-AI)
Representation (weeks 4-6)

- base-N representations
- images and compression
- logic gates: AND OR NOT
- circuit composition
- assembly programming

CS (non-AI)

AI

Audacity

audio
Media computation: both sides

expressive creativity

def reverse( sound ):

def oneFreq( freq ):

def myeffect( ... ):

creating audio data
Media computation: both sides

expressive creativity  analytic creativity

def dft( sound, freq ):
    sinWave = oneFreq( freq )
    power = dot( sound, sinWave )
    return power
Media computation: both sides

---

**synthetic creativity**  ↔  **analytic creativity**

---

```python
def dft( sound, freq ):
    sinWave = oneFreq( freq )
    power = dot( sound, sinWave )
    return power
```

---

What mood
are you in?

---

and **breaking** it apart: chord classification *for any key*
'Cause somethin' like he left knee and a harp," said he had to the whole school? The shouting and then some strange and Mrs. "Well, I know Hagrid; they spotted handkerchief and get him get rid of course, had a gigantic beet with her," he knew what to all he's

All the sky with the sun in the sun in the church where you're gone Lucy in my eyes. There beneath the girl with an hourglass And then the banker never wears a lot to hold your hand. Can't buy me tight, tight Owww! Love is love I can't hide,

This is but ourselves. No, faith, My uncle! O royal bed of confession Of your rue for leave to nature; to this time I should weep for thy life is rotten before he is. have sworn 't. Or my blood. I have closely sent for nine; and unprofitable,
Imperative

'Cause somethin' like he left the harp," said he had to the whole.

The shouting and then some Mrs. "Well, I know Hagrid; handkerchief and get him get course, had a gigantic beet what he knew what to all he's

...ky with the sun in the sun in the where you're gone Lucy in my beneath the girl with an And then the banker never got to hold your hand. Can't buy tight Owww! Love is love I

This My uncle! O rue for

Rooter: A Methodology for the Typical Unification of Access Points and Redundancy

Jeremy Stribling, Daniel Aguayo and Maxwell Krohn
Other work \textit{weeks 7-9}

- Life
- Gaussian elimination
- Physical time-step simulations
- Monte Carlo simulations
- Flesch readability
- ASCII art
- Mandelbrot Set
class Board: ...

b = Board(6,7)
b.hostGame()

| | | | | | | |
| | | | | | | |
| | | | | | | |
|O| |O| | | | |
|O| |X| | |X|X|
|X|X|X|X|O|O|O|
---------------
0 1 2 3 4 5 6

X wins!

class Player:...

b = Board(6,7)
pX = Player('X',2)
pO = Player('O',2)
b.playGame(pX,pO)

|O|O|X|O|O|O| |
|X|X|O|X|X|X| |
|O|O|X|O|O|O| |
|X|X|O|X|X|X| |
|O|O|X|O|O|O|O|
|X|X|X|O|X|X|X|
---------------
0 1 2 3 4 5 6

O wins!
class Board: ...

b = Board(6,7)

class Player:...

b = Board(6,7)
pX = Player('X',2)
pO = Player('O',2)

BWAHAHAHAHAHAHAHAHAHAHAH - I AM THE WINNER AND IT ALL WORKS!!!
GWAHAHAHAHAHAHAHAHAHAH!!  I started at noon, I'm done at 11:11 (minus a brief break to watch shawshank redemption and eat dinner, I've been working solid ;) )  I AM TOTALLY THE WINNER!

X wins!  O wins!
implementing finite-state machines for a robot-inspired navigation task
Theory & Projects  weeks 12-14

Take 1

Take 2
What do students get out of this experience?

Breadth  Enthusiasm  Retention
Results

What is something a CS researcher might study?

- **Algorithmic efficiency**
- **AI/Robotics/Vision**
- **Programming**
- **Simulations/Graphics**
- **Storage/Networking**
- **Hardware**
- **Math/Cryptography**
- **New Languages**
- **How to help society**
- **Other topics**
- **Computability**

*After* and *Before* responses for each topic.
Results

Students reported working more in CS 1 than other intro courses…

And felt that the work would pay off.

Core curriculum students’ hours per week (and anticipated grades), Fall ’06

<table>
<thead>
<tr>
<th></th>
<th>Physics</th>
<th>CS 1</th>
<th>Math</th>
<th>Writing</th>
<th>Bio</th>
<th>Engin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Women</td>
<td>5.2 (C+)</td>
<td>7.2 (B)</td>
<td>6.4 (B-)</td>
<td>5.3 (B-)</td>
<td>2.8 (B-)</td>
<td>7.0 (B-)</td>
</tr>
<tr>
<td>Men</td>
<td>5.9 (B-)</td>
<td>7.3 (B+)</td>
<td>6.6 (B)</td>
<td>5.7 (B-)</td>
<td>2.3 (B)</td>
<td>6.2 (B)</td>
</tr>
</tbody>
</table>
# Results

**Enthusiasm & breadth**

A) The course stimulated my interest in the subject matter.

B) I learned a great deal in this course.

<table>
<thead>
<tr>
<th>Students Considered</th>
<th>Agreement with (A), from 1-7</th>
<th>Agreement with (B), from 1-7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall 2005: all courses</td>
<td>5.65</td>
<td>5.71</td>
</tr>
<tr>
<td>Fall 2005: traditional CS1</td>
<td>5.14</td>
<td>5.81</td>
</tr>
<tr>
<td>Fall 2006: CS for Scientists</td>
<td>5.89</td>
<td>6.11</td>
</tr>
<tr>
<td>Fall 2006: all courses</td>
<td>5.70</td>
<td>5.80</td>
</tr>
</tbody>
</table>

on a seven-point scale
Results

Student-reported worthwhileness and difficulty

Likert score from 1 (least) to 7 (most)
Results

Student-reported worthwhileness and difficulty

Worthwhile?
Results

Student-reported worthwhileness and difficulty

Difficult?

0.83 correlation
Results

Retention flat…
but more women!

Students choosing to take CS 2 in the spring

- **2003**: old CS 1 (19.7), new CS 1 (6.9)
- **2004**: old CS 1 (19.4), new CS 1 (6.8)
- **2005**: old CS 1 (21.4), new CS 1 (5.6)
- **2006**: old CS 1 (23.0), new CS 1 (7.0)
- **2007**: old CS 1 (15.3), new CS 1 (13.4)
Results

Comparison between women's and men's responses to AI-themed CS 1 assignments

- **Difficulty**, reported by women:
  - Markov Text: 5.25
  - C4 Board class: 4.75
  - C4 Player class: 5.75
  - Robotics: 6.25
  - Caesar Cipher: 5.25
  - Sound: 4.75

- **Difficulty**, reported by men:
  - Markov Text: 4.25
  - C4 Board class: 3.75
  - C4 Player class: 4.75
  - Robotics: 5.25
  - Caesar Cipher: 5.75
  - Sound: 4.25

- **Worthwhileness**, reported by women:
  - Markov Text: 5.75
  - C4 Board class: 5.25
  - C4 Player class: 6.25
  - Robotics: 5.75
  - Caesar Cipher: 5.25
  - Sound: 4.75

- **Worthwhileness**, reported by men:
  - Markov Text: 4.75
  - C4 Board class: 4.25
  - C4 Player class: 5.25
  - Robotics: 5.75
  - Caesar Cipher: 5.25
  - Sound: 4.75

*only significant differences: women reporting the hw more difficult*

\[ t = 0.05 \]
Beyond the numbers

Six-week summer project for three first-year women
Verdict

Breadth ↑ Enthusiasm ↑ Retention ↔

AI ~ motivating without dominating.

politically flexible!  CS 1 is where we faculty are now most passionate.

disproportionate impact: Cultivating the Long Tail

www.cs.hmc.edu/twiki/bin/view/CS5/WebHome
In CS 2

Comparing old vs. new CS 1 students, in terms of performance in CS 2
( cohort numbers )

<table>
<thead>
<tr>
<th>CS 2, Spring '07</th>
<th>Midterm exam scores, %</th>
<th>Final exam scores, %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>women</td>
<td>men</td>
</tr>
<tr>
<td>Java CS 1</td>
<td>84.0</td>
<td>79.5</td>
</tr>
<tr>
<td>Python CS 1</td>
<td>80.7</td>
<td>84.7</td>
</tr>
</tbody>
</table>

85% chance of significance

Starting with Python was no disadvantage, even without changing later courses.
Results

Comparison of students' perceived importance of CS

Comparison of students' perceived importance of CS before and after the intervention.
Final exam  week 15

Old exam median: 85%

New exam median: 77%

Median new-exam scores

<table>
<thead>
<tr>
<th>New CS 1</th>
<th>All</th>
<th>Women</th>
<th>Men</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam</td>
<td>76.9</td>
<td>71.5</td>
<td>79.0</td>
</tr>
<tr>
<td>Full course</td>
<td>89.9</td>
<td>89.6</td>
<td>90.5</td>
</tr>
<tr>
<td>Homework</td>
<td>92.3</td>
<td>92.8</td>
<td>92.3</td>
</tr>
</tbody>
</table>
Old and new exam questions

1. Create and compose three small functions to spec.
2. Demonstrate correct use of references vs. raw data
3. Manipulate a 1d data structure, e.g., mode
4. Write a program searching a 2d array along both dims.
5. Design/write a Time class with data, methods to spec.

Additional new exam questions

7. Compose an assembly-language routine to spec.
8. Prove a (previously unseen) function uncomputable
The 17th Annual AAAI Robot Workshop and Exhibition

Chicago, IL, USA, July 13-17, 2008

Tracks:

1. Robotics and Creativity
2. Mobility and Manipulation

Email Paul Oh this week!

Theme: Teaching AI per se

Submissions of 2-6 pages by April 7, 2008.
AAAI '08 Opportunities
CS 1 options!

**Precedence**

- imperative-first
- objects-first
- functional-first
- breadth-first
- algorithms-first
- hardware-first
- rigor-first

**Thematic Structure**

- robots
- games
- media
- math & science
CS 1 options!

Precedence

- imperative-first
- objects-first
- functional-first
- breadth-first
- algorithms-first
- hardware-first
- rigor-first

Thematic Structure

- robots
- games
- media
- math & science
Functional Finale

Caesar deciphering

An education is what remains when we forget everything we have learned.

Et tu, Brute?

CS composition, design, map, and reduce (or list comprehensions)

solution

wonderfully Al-ish!

run forward and optimize
Verdict

AI can be motivating without dominating.

offers challenge - requires creativity

disproportionate impact: cultivating the long tail

A dozen first-year women are working with our CS department this coming summer

politically flexible !

CS 1 is where we are now most passionate.

What is learned is the square root of what is taught.

www.cs.hmc.edu/twiki/bin/view/CS5/WebHome
### Results

#### Motivation?

Students reported working more in CS 1 than other intro courses... And felt that the work would pay off.

<table>
<thead>
<tr>
<th></th>
<th>Physics</th>
<th>CS 1</th>
<th>Math</th>
<th>Writing</th>
<th>Bio</th>
<th>Engin.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Women</td>
<td>5.2 (C+)</td>
<td>7.2 (B)</td>
<td>6.4 (B-)</td>
<td>5.3 (B-)</td>
<td>2.8 (B-)</td>
<td>7.0 (B-)</td>
</tr>
<tr>
<td>Men</td>
<td>5.9 (B-)</td>
<td>7.3 (B+)</td>
<td>6.6 (B)</td>
<td>5.7 (B-)</td>
<td>2.3 (B)</td>
<td>6.2 (B)</td>
</tr>
</tbody>
</table>

Core curriculum students’ hours per week (and anticipated grades), Fall ’06
Verdict

Primum non nocere.

CS 1 is where we are now most passionate.

www.cs.hmc.edu/twiki/bin/view/CS5/WebHome
You take one down, pass it around,
97 bottles of beer on the wall!
97 bottles of beer!
You take one down, pass it around,
96 bottles of beer on the wall!
96 bottles of beer!
You take one down, pass it around,
95 bottles of beer on the wall!
95 bottles of beer!
You take one down, pass it around,
94 bottles of beer on the wall!
94 bottles of beer!
You take one down, pass it around,
93 bottles of beer on the wall!
93 bottles of beer!
You take one down, pass it around,
92 bottles of beer on the wall!
92 bottles of beer!
You take one down, pass it around,
91 bottles of beer on the wall!

Traceback (most recent call last):
  File "<pyshell#3>", line 1, in _toplevel_
bottlesFallSong()
  File ""/private/var/run/wst/app/courses/gradc/gradc.py", line 6, in bottlesFallSong
    raise ValueError("Failed to load bottlesFallSong()
Failed to load bottlesFallSong()
2) Create a circuit for segment 'g'

5) Floating-point division in (integer-only) assembly

6) Count words in a 2d array

   >>> wordCount( 'spam', [ 'asmic', 'spams', 'papaj', 'amsoy' ] )
   3

8) Show that contrary is uncomputable

   def contrary( prog ):
       """ contrary takes in a python function, prog
       prog will always be a python function of zero inputs

       contrary returns True if prog() returns False and
       contrary returns False otherwise
       """
Perspective

Keep

- breadth-first
- closed labs and final project
- homework choices

Change

- imperative with functional
- 8 am class time
- Audio vs. 3d

www.cs.hmc.edu/twiki/bin/view/CS5/WebHome
the breadth-first model has not enjoyed the success that its proponents had envisioned… most breadth-first courses that exist today seem to be lead-ins to a more traditional programming sequence

- CC 2001
Final exam

Old exam median: 85%

New exam median: 77%

Median new-exam scores

<table>
<thead>
<tr>
<th>New CS 1</th>
<th>All</th>
<th>Women</th>
<th>Men</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final exam</td>
<td>76.9</td>
<td>71.5</td>
<td>79.0</td>
</tr>
<tr>
<td>Full course</td>
<td>89.9</td>
<td>89.6</td>
<td>90.5</td>
</tr>
<tr>
<td>Homework</td>
<td>92.3</td>
<td>92.8</td>
<td>92.3</td>
</tr>
</tbody>
</table>
Objects  weeks 10-11

emphasizing use over design

Date

def dow(self):
    dayOfWeekList = ["Thu", "Fri", "Sat", "Sun", "Mon", "Tue", "Wed"]
    now = Date(3,27,2008)
    b = self.diff(now)
    return dayOfWeekList[b%7]

at the command line:

(30)% python -i hw11pr1.py
>>> d = Date(3,4,2009)
>>> d.dow()
Wed

an object-based Date calculator

ASCII Connect 4

p1 = Player('X',2)
p2 = Player('O',0)
b = Board(6,7)
b.playGame(p1,p2)

| | | | | | | |
| | | | | | | |
| | | | | | | |
|O| |O| | | | |
|O| |X| | |X|X|
|X|X|X|X|O|O|O|
---------------
0 1 2 3 4 5 6

X wins!

Connect Four with lookahead
CS 1 at HMC

traditional, mostly imperative, Java

<table>
<thead>
<tr>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Choose a light: 2

<table>
<thead>
<tr>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Choose a light: 1

<table>
<thead>
<tr>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
<th>****</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Choose a light: 0

You win!

lights out

'mystery' function

f("onyx","balk") == 13.0
f("adds","beet") == 1.0
f("zach","bach") == 0.5

abstract(ion) art

wk 1-2: variables
wk 3-4: control
wk 5-6: functions
wk 8-9: arrays
wk 10+: objects
the breadth-first model has not enjoyed the success that its proponents had envisioned… most breadth-first courses that exist today seem to be lead-ins to a more traditional programming sequence

- CC 2001
the breadth-first model has not enjoyed the success that its proponents had envisioned… most breadth-first courses that exist today seem to be lead-ins to a more traditional programming sequence

- CC 2001

Goals:

- develop computational skills sufficient for CS 2 and useful for any scientific field of study
- draw more students, especially women, to CS
- build a context for CS: an important field of study
CS 1 for scientists

2-3 lectures per week, 1 lab
two experience-based sections

CS breadth

<table>
<thead>
<tr>
<th>Weeks</th>
<th>Paradigm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>functional</td>
</tr>
<tr>
<td>4-6</td>
<td>machine-level</td>
</tr>
<tr>
<td>7-9</td>
<td>imperative</td>
</tr>
<tr>
<td>10-12</td>
<td>objects+classes</td>
</tr>
<tr>
<td>13-15</td>
<td>theory/projects</td>
</tr>
</tbody>
</table>

AI-themed labs and assignments

integration, random walks, ciphers
recursion in assembly, 4-bit multiplier
Markov text generation, Conway's life
Connect Four player, Date calculator
uncomputability, finite-state machines

AI breadth