Welcome to IST338!

Grab these lecture notes...

CS for Insight

Wally Wart, a protrusive advocate of *concrete* computing

We don't have words strong enough to describe this class.
- US News and Course Report

Everyone will get out of this course – a lot!
- NYTimes Review of Courses

We give this course two thumbs...
- Metametacritic

1 handout...
slides & syllabus
Introductions...

Zach Dodds
HMC Olin B163
dodds@cs.hmc.edu

pursuer of low-level AI

taker of low-quality photos

fan of low-tech games

Speaking of introductions
Introductions...

Zach Dodds
HMC Olin B163
dodds@cs.hmc.edu

pursuer of low-level AI
taker of low-quality photos
fan of low-tech games

Speaking of introductions
How I spend my breaks...?

actually, this "I" is not quite accurate... 🤗

Robots

Chairs?

Hawaii?
“Quiz”
introductions

- Name
- Preferred email
- A place you consider home other than Claremont.
- Your favorite __________ is _____________.
- Your least favorite _____________ is _____________.
- Are you working at a company now? If so, where?
- Do you use computer programming? How much/what languages?

May I take a quick movie of you introducing yourself?
“Quiz”

- Name: Zachary Dodds
- Preferred email: dodds@cs.hmc.edu
- A place you consider home: Pittsburgh, PA
- Your favorite TV show is: Dr. Who
- Your least favorite coffee is: decaffeinated
- Are you working at a company now? Where? Harvey Mudd
- Do you use computer programming? How/what? python/others...
A word on 5 spots...

negotiations are happening here… but don't affect CGU at all…
Today in 338...

2) How the course runs...

3) Python?!

1) Why CS? & What's CS?

Whatever it is, it's definitely alien!

CS is just programming, right?

I'm not so sure...

Shouldn't there be an alien in this game?
Why CS? Who's studying CS?

Every first-year student at Mudd must.

Right now, 200 students from the other colleges are!

This means there is lots of support ...

This class, naturally, takes a big-picture view

• less emphasis on programming for its own sake
• more emphasis on the context and the mindset!
CS vs. IS and IT?

**Figure 2.2. The problem space of computing**

- **greater integration**
- **system-wide issues**

- **smaller details**
- **machine specifics**

CS vs. IS and IT?

2.3.2. Computer Science

The shaded portion in Figure 2.4 represents computer science. Computer science covers most of the vertical space between the extreme top and extreme bottom because computer scientists generally do not deal with “just the hardware” that runs software, or about “just the organization” that make use of the information that computing can provide. As a group, computer scientists care about almost everything in between those areas (down as far as the software that enables devices to work; up as far as the information systems that help organizations operate). They design and develop all types of software, from systems infrastructure (operating systems, communications programs, etc.) to application technologies (web browsers, databases, search engines, etc.) Computer scientists create these capabilities, but they do not manage the deployment of them. Therefore, the shaded area for computer science narrows and then stops as we move to the right. This is because computer scientists do not help people to select computing products, nor tailor products to organizational needs, nor learn to use such products.

Where will IS go?

Figure 2.4. Computer Science

Figure 2.5. Information Systems
2.4.2. Computer Science

The shaded portion in Figure 2.4 represents computer science. Computer science covers most of the vertical space between the extreme top and extreme bottom because computer scientists generally do not deal with “just the hardware” that runs software, or about “just the organization” that make use of the information that computing can provide. As a group, computer scientists care about almost everything in between those areas (down as far as the software that enables devices to work; up as far as the information systems that help organizations operate). They design and develop all types of software, from systems infrastructure (operating systems, communications programs, etc.) to application technologies (web browsers, databases, search engines, etc.) Computer scientists create these capabilities, but they do not manage the deployment of them. Therefore, the shaded area for computer science narrows and then stops as we move to the right. This is because computer scientists do not help people to select computing products, nor tailor products to organizational needs, nor learn to use such products.

2.4.3. Information Systems

The shaded portion in Figure 2.5 represents the discipline of information systems. The shaded area extends across most of the top-most level because IS people are concerned with the relationship between information systems and the organizations that they serve, extending from theory and principles to application and development; many IS professionals are also involved in system deployment and configuration and training users. The area covered by IS dips downward, all the way through software development and systems infrastructure in the right half of the graph. This is because IS specialists often tailor application technologies (especially databases) to the needs of the enterprise, and they often develop systems that utilize other software products to suit their organizations’ needs for information. (This figure does not reflect the attention that information systems programs devote to core business topics. See Chapter 3 for tables which summarize both computing and non-computing topics.)
2.3.2. Computer Science

The shaded portion in Figure 2.4 represents computer science. Computer science covers most of the vertical space between the extreme top and extreme bottom because computer scientists generally do not deal with "just the hardware" that runs software, or about "just the organization" that make use of the information that computing can provide. As a group, computer scientists care about almost everything in between those areas (down as far as the software that enables devices to work; up as far as the information systems that help organizations operate). They design and develop all types of software, from systems infrastructure (operating systems, communications programs, etc.) to application technologies (web browsers, databases, search engines, etc.) Computer scientists create these capabilities, but they do not manage the deployment of them. Therefore, the shaded area for computer science narrows and then stops as we move to the right. This is because computer scientists do not help people to select computing products, nor tailor products to organizational needs, nor learn to use such products.

IT?

Where would IT go?

Figure 2.4. Computer Science

Figure 2.6. Information Technology
2.4.4 Information Technology

The shaded portion in Figure 2.6 represents the information technology discipline. Its shaded area extends down most of the right edge, as it focuses on the application, deployment, and configuration needs of organizations and people over a wide spectrum. Across this range (from organizational information systems, to application technologies, and down to systems infrastructure), their role has some overlap with IS, but IT people have a special focus on satisfying human needs that arise from computing technology. In addition, IT's shaded area goes leftwards, from application towards theory and innovation, especially in the area of application technologies. This is because IT people often develop the web-enabled digital technologies that organizations use for a broad mix of informational purposes, and this implies an appropriate conceptual foundation in relevant principles and theory.

Because IT is a very new discipline, its focus has been on developing educational programs that give students a foundation in existing concepts and skills. Many in the community of IT faculty assert that research in their field will grow to create and develop new knowledge in relevant areas. When this happens, an appropriate snapshot would feature a shaded area that extends significantly further to the left. However, this is an ambition and not yet an achievement. This figure reflects IT's current status.

What specialty lies in the "lower" portion of these axes?
Who else is studying CS?
Who else is studying CS?

Computer science enrollments rocketed last year, up 22%

Demand for technical workers lures B.S. candidates in CompSci

By Bob Brown | Follow
NetworkWorld | Mar 11, 2014 11:14 AM PT

Many people...
Who else is studying CS?

Many people at some places...
Many companies want a larger diversity of skillsets, mindsets, and talents — but all of them empowered by CS
CS? *Leap* before you look!

**Browser Python!**

Being 2015, you can run Python entirely from your browser: the example below is from Trinket, which seems the most capable as of right now... .

Chrome, Safari, Firefox -- all should work (maybe IE?). Also, you will be able to write all of the Python you need - at least for hw0 - using Trinket. Some of the later assignments will require a Python installed on your own machine.

hw0pr0: Run and submit this program...  
(extra) Write a rock-paper-scissors program...
Get to the CS5 website and hw0. Complete hw0pr0 - in brief,

1. Make sure you can log into the submission system
   
   username = first initial + last name (most often)
   password =

   troubles? email me!

2. Change your password + upload an image
3. Run and submit $\text{hw0pr0.py}$ (provided!)

$Extra$: submit a rock-paper-scissors-playing program...

$Extra! Extra!$ ~ if you do the extra credit this week, your first problem for next week is already complete!
This week's hw... by next Fri. night:

Get to the CS5 website and hw0. Complete hw0pr0 - in brief,

1. Make sure you can log into the system
   
   username = [first initial + last name] (most often)
   password = [troubles?
email me!]

2. Change your password + upload an image

3. Run and submit hw0pr0.py (provided!)

Extra: submit a rock-paper-scissors-playing program...

Extra! Extra! ~ if you do the extra credit this week, your first problem for next week is already complete!
But, what is CS a science of?

the study of complexity:

How can it be done?
How well can it be done?
Can it be done at all?

it ~ information

or, more precisely, a process transforming information from one form to another
What is CS a science of?

the study of *complexity*:

*How can it be done?*

*How well can it be done?*

*Can it be done at all?*

*it ~ information*

or, more precisely, *a process transforming information from one form to another*

We'll look at 3 examples – each of which you'll **build** in IST380e...

...at least to some extent!
What is CS?

Can you solve the problem?
Can you create a process to solve such problems?

'HUMAN'

'CHIMPANZEE'

What is the Longest Common Subsequence between 2 strings?
biology's string-matching problem, "LCS"

'CGCTGAGCTAGGCA...'

'ATCCTAGGTAACTG...'

+10^9 more

Eye oneder if this haz othur aplications?
What is CS?

How can it be done?
How well can it be done?
Can it be done at all?

How quickly can you find a solution?
Is your solution the "best" possible?

How much work is needed to simulate N stars?

physics's "N-body" problem

what if N is a billion-and-one...?
What is CS?

How can it be done?
How well can it be done?
Can it be done at all?

Is your problem solvable?
How can you tell !?

many problems are uncomputable...
... and you'll prove this!

Andrew Ng's "Make3d"

Can we build a 3d model from one 2d image?

All three eyes tell me that Make3d has just failed ~ epically!
What is CS?

CS is the study of *complexity*

How can *it* be done?

How well can *it* be done?

Can *it* be done at all?

Can you solve this problem?

Can you create a process to solve such problems?

How quickly can you find solutions?

Do you have the “best” solution?

Is every problem solvable?

Is there a way to tell?

There isn’t always!

But only *one* is *programming*.

Do you see which?
What is CS?

CS is the study of **complexity**

**How can it be done?**

**How well can it be done?**

**Can it be done at all?**

CS's 6 big questions are here.

But only **one is programming**.

Do you see which?

---

Can you solve this problem?

Can you create a process to solve such problems?

How quickly can you find solutions?

Do you have the “best” solution?

Is every problem solvable?

Is there a way to tell?

There isn’t always!
CS’s philosophy:

Whatever you are, be a good one.

- Abraham Lincoln

More and more, CS can help!
If Silicon Valley - or Silicon Beach – opened a fn. cookie...

You would do well in the field of computer technology.

Soundbite Syllabus

**Lectures**
- **W 7-9:50** (MW: 1:15 – 2:30 pm) HMC
  - Key ideas, proof, and the intuition behind them
  - Insight into the HW problems (what, why, how)
  - I'd like to see you! Let me know if you'll be sick...

**Lab**
- **W 9-9:50** (M or T: 2:45 - 4:45 pm or 8:00 - 10:00 pm) HMC
  - Guided progress on the class’s hw
  - Not required, but recommended by 4 out of 5 CS-for-Insight alums!
  - Will help save you time and effort to get started!

**Office hrs**
- **W 3-5pm** Olin B163
  - Come to Lab!
  - Come to Lectures!
  - Lots of help is available!

**HW**
- **Friday HW**
  - Hw is due on Sunday nights...
Syllabus, briefly

**Lectures**

- **W 7-9:50** (MW: 1:15 – 2:30 pm)  **HMC**
  - Key skills, topics, and their motivation
  - Insight into the HW problems (what, *why*, how)
  - I'd like to see you! Let me know if you'll be sick...

**Lab**

- **W 9-9:50** (M or T: 2:45 - 4:45 pm or 8:00 - 10:00 pm)  **HMC**
  - Guided progress on the class's hw
  - Not required, but encouraged
  - Will *SAVE* you time and effort to get started!

**Office hrs**

- **W: 3-5pm  Olin B163**  **same for HMC**
  - feel free to work on HW, to just stop by,
  - or, come to any of the *many* tutoring hrs!

**HW**

- **Friday nights:** due by 11:59 pm - ish
Homework

Assignments  ~ 5 problems/week

Due Friday evenings by 11:59 pm.

Extra credit is usually available...

You have 3 CS 5 Euros to use...
"Late Days"

Collaborate!

Some problems are specified “individual-only.”
Others offer the option of working as pairs/partners:

• You don’t have to work in pairs/partners (that said, it's fun!)
• You must share the work equally - typing and coaching
• Be sure to indicate who your partner was at the submission site!
• Put your name(s) in the code, as well!

Eur-ollowed to use one Euro for up to three hwks.

No need to let us know, even.
Pairs

one computer
tradeoff typing/debugging ~ about every 20 minutes

Partners

two computers
both partners type/debug ~ provide help as needed

Standard is the same either way:
After finishing the hw, (a) each person has contributed equally and (b) both could complete the problems on their own
your work matters

• You're encouraged to discuss problems with other students – or tutors - or any instructors.

• You may not share written, electronic or verbal solutions with other students, present or past:

Please do use the internet for Python language references.

Please do use other's eyes for finding syntax errors.

Do not use the internet (or intranet) to (try to) find solutions...

If you work as a pair/partners, the rules apply for the group.

Claremont-wide code: Submit only your own work.
Grading

~ 65% Assignments
~ 30% Final project
~ 5% Participation/“quizzes”

There are no exams in IST338...

... there is a final project, however.
Suppose that the value of \texttt{perc} is 0.91...

\begin{verbatim}
perc = 0.91

if perc > 0.95:
    print 'A'
elif perc > 0.90:
    print 'A-
elif perc > 0.70:
    print 'Pass'
else:
    print 'Aargh!'
\end{verbatim}

What will this program print, if \texttt{perc} is 0.91?
Choices, choices!

perc = 0.80

if perc > 0.95:
    print 'A'
elif perc > 0.90:
    print 'A-
elif perc > 0.70:
    print 'Pass'
else:
    print 'Aargh!'

perc = 0.80

if perc > 0.00:
    print 'Aargh!
elif perc > 0.70:
    print 'Pass'
elif perc > 0.90:
    print 'A-
else:
    print 'A'

What does each of these programs print out, if \texttt{perc} is 0.8?

What value of \texttt{perc} gives an \texttt{'A-'} on the right?

How can you get a better grade on the right than the left?
Exclusive Choices

if ... elif ... else

```python
if perc > .95:
    print 'A'

elif perc > .90:
    print 'A-

elif perc > .70:
    print 'Pass'

else:
    print 'Aargh!'```

4 mutually exclusive blocks
in a single control structure

When using
if . elif ... . else
at most one block will run:
the first whose test is True.
If all fail, the else will run

elif and else are optional
What's the difference?

mutually exclusive blocks

```python
if perc > .95:
    print 'A'
elif perc > .90:
    print 'A-
elif perc > .70:
    print 'Pass'
```

non-exclusive blocks

```python
if perc > .95:
    print 'A'
if perc > .90:
    print 'A-
if perc > .70:
    print 'Pass'
```

What if `perc == .99`?

How many separate control structures does each side have?
Next time in IST338

1) What is CS?

Whatever it is, it's definitely alien!

2) Python!

I'm not so sure…

Shouldn't there be an alien in this game?
*hw0pr0 – getting started*

*due a week from Fri, 1/30...*

Thank you, everyone!!

How about a sneak peek at next week?

... you must mean sneak *Pic*!