

Wally Wart, a protrusive advocate of **concrete** computing

Welcome to CS 5 !

Grab these lecture notes...

Introduction to CS

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

1 handout...
slides & syllabus

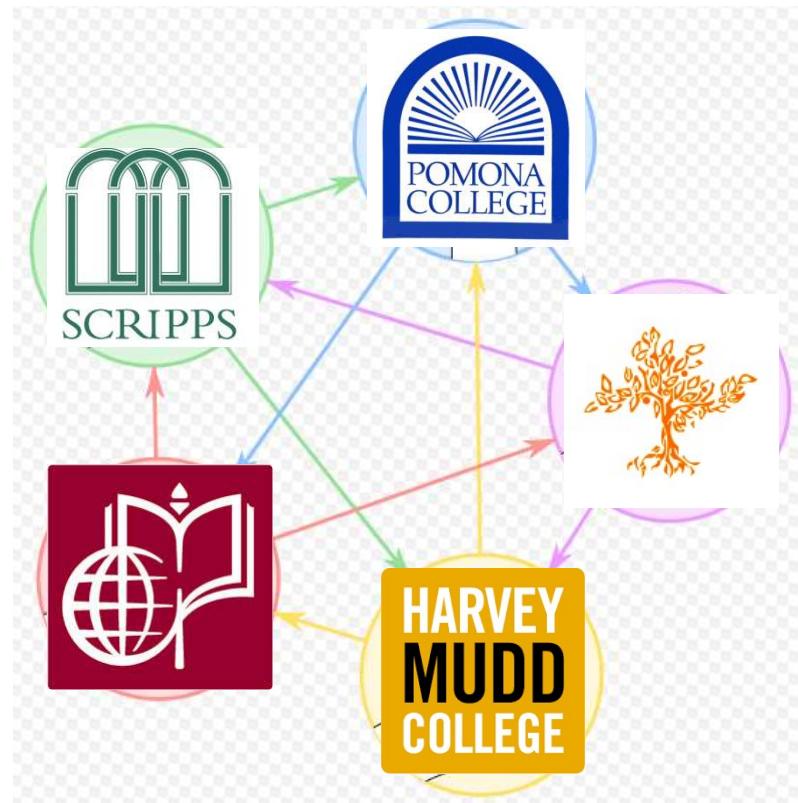
We give this course two thumbs...

- Metacritic



official alien of CS 5 Gold

A word on 5 spots...



negotiations afoot... *this section may fill*

Introductions...



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

pursuer of *low-level AI* →

↖ taker of *low-quality selfies*
fan of *low-tech games* ↓

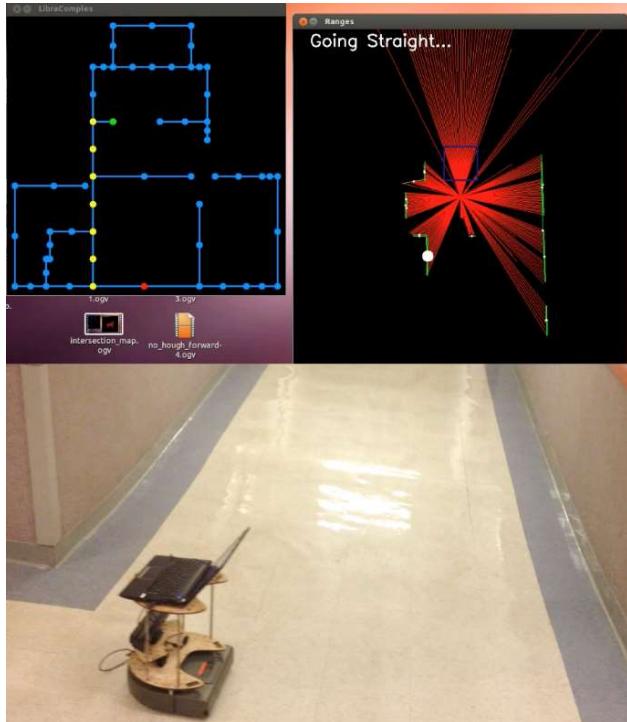


Speaking of
introductions



How I spend my summers ...?

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



Robots



Chairs?



Outreach

Algorithmic improvisation



Start-up ideas...



... to formal pitches



CS Staff: rising sophomores, unite!



CS Staff: Rising sophomores, unite!



Lots of opportunities
surrounding computing...
(at the 5Cs and beyond)

Today in CS5

2) How CS 5 runs...

3) Python?!



1) What *is* CS?



CS is just programming, right?

Whatever it is,
it's definitely *alien*!



I'm not so sure...



CS vs. programming ?

A minute of cs5 programming...

Python source code,
a plain-text file

(here, edited by the VS Code text editor)

lab and hw
instructions

Python source code,
a plain-text file
(here, edited by the VS Code text editor)

• Running a file!

To run your file, go back over to the terminal.

- Type ipython if you're not yet running it.
- Type ls (windows or mac) to see the files in the current directory
- Make sure your hw0pr1.py file is there!
 - If not, use cd .. or cd Desktop or other combinations to get to the correct directory. Ask for help!
- At the ipython prompt, type run hw0pr1 (tab completion will work)
- This should run the file hw0pr1.py
- If all goes well, the program should run and you should see the output
- If not, please ask!
- Now, you can edit your file, save it, and b...

Your task: four fours challenge

- The **four fours challenge** asks you to find the 21 values from 1 to 21 that can be created using the four basic operations:
 - + addition
 - - subtraction
 - * multiplication
 - / division
 - () parentheses
 - ** power
- You may also use .44 or .444, which counts as one digit.
- See below for two more advanced challenges.
- Hint: There are many ways to look at the results, but you need only one way to look like.

Demo

shell or command-line or terminal
(the execution environment)

get everything running *on your own machine*

Spot the difference here?

```
print('hi')
```

```
print 'hi'
```

I could not – *for days!*

Spot the difference here?

```
print('hi')
```

python 3

```
print 'hi'
```

python 2

We will be using python 3 this term...

Spot the difference here?

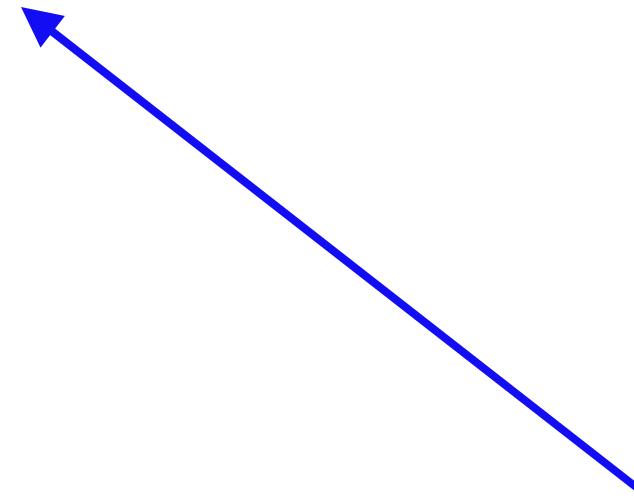
```
print('hi')
```

```
print 'hi'
```

Syntax!

We will be using python 3 this term...

CS != programming



"not equal to"

CS != programming

programming : CS ::

longboards : HMC maybe 5Cs?

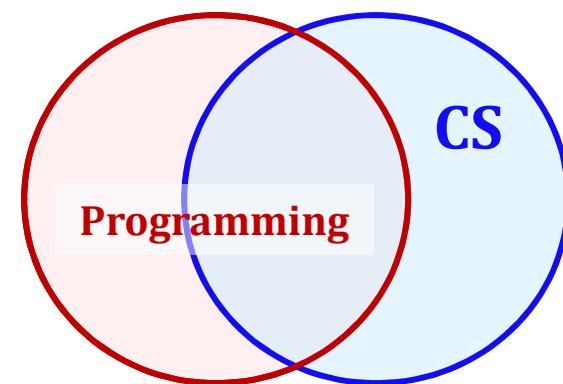
capital : business venture

equations : mathematics

language : ideas

web search : knowledge

Tesla : Google



programs are a *vehicle*, not the destination

CS != programming

So, what is CS?

Punctuation matters!
So what? *is* CS



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?*

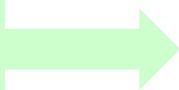
it ~ information

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



3 examples?
That's **it** for me!

We'll look at 3 examples – each of
which you'll **construct** in CS 5
...at least to some extent!



What is CS?

'HUMAN'

'CHIMPANZEE'

*How can **it** be done?*

*How well can **it** be done?*

*Can **it** be done at all?*

What is the *Longest Common Subsequence* between 2 strings?

biology's string-matching problem, "LCS"

'CGCTGAGCTAGGCC...'

'ATCCTAGGTAACTG...'

+10⁹ more

Can you solve the problem?

Can you create a *process* to solve such problems?

Eye oneder if this haz
othur applications?



Feels like home!



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?

The screenshot shows the homepage of the **universe SANDBOX** website, created by Dan Dixon. The page features a 3D simulation of celestial bodies, including planets and stars, against a dark background. At the top right, there are links for "what is it?", "download", "buy", "forum", and "contact". Below these are links for "videos", "screenshots", and "faq". A prominent button says "Install Universe Sandbox" with the note "Requires Windows XP, Vista, or Windows 7". To the right, there is a video player titled "Watch the Video" showing a simulation of a star system. The video player includes controls for play, volume, and full screen, along with a timestamp of "0:38 / 1:15". Below the video player are links for "View the YouTube page" and "Watch in Larger Window".

*How much work is needed
to simulate N stars?*

chemistry's + physics's " N -body" problem

what if N is a million-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!



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

Andrew Ng's "Make3d"

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?*

CS's **6** big
questions

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!*

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

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 – and CS5's – philosophy:

*Whatever you are,
be a good one.*

- Abraham Lincoln

More and more,
CS can help!

Take-home message...

depending on where
"home" is, perhaps...



CS5 - WebHome

https://www.cs.hmc.edu/twiki/bin/view/CS5

Apps CS5

CS 5: Welcome!

Administration	Course Syllabus	Exams	Pairs Policy	Submission Info
Using Python	On your machine	In your browser	CS5's text	httlacs text
Useful/Help	Submission site	CS5 Piazza	Grutoring!	Picobot

Homework Assignments and Labs

Labs	GOLD	BLACK
Week 0	Lab 0	Homework 0

Lecture Slides

(Before class, the previous term's slides might be posted; shortly after class the current slides will replace them.)

	GOLD	BLACK
Week 0		
10/30/16	Lecture 0: Introduction	Lecture 0: Introduction
9/1/16	Lecture 1: Pico-fun!	Lecture 1: Map and Reduce

Acknowledgments and thanks...

*Yay! in 2017
Just Google for
hmc cs5*

www.cs.hmc.edu/cs5

You're here ~ what's next?

2) How CS 5 runs...

3) Python?!

the first Python HW
is *choice!*



1) What *is* CS?

Whatever it is,
it's definitely *alien!*

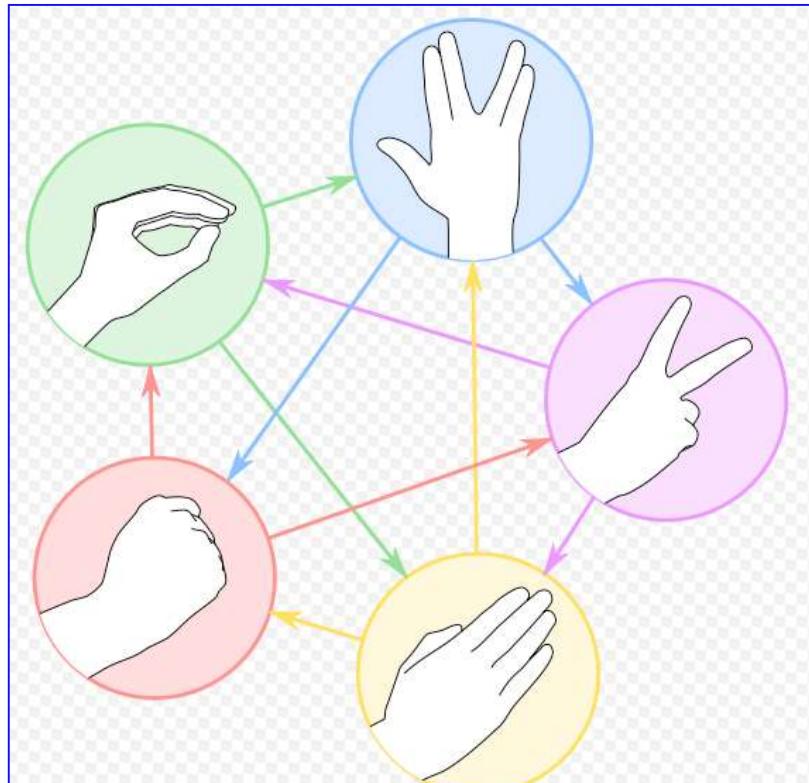
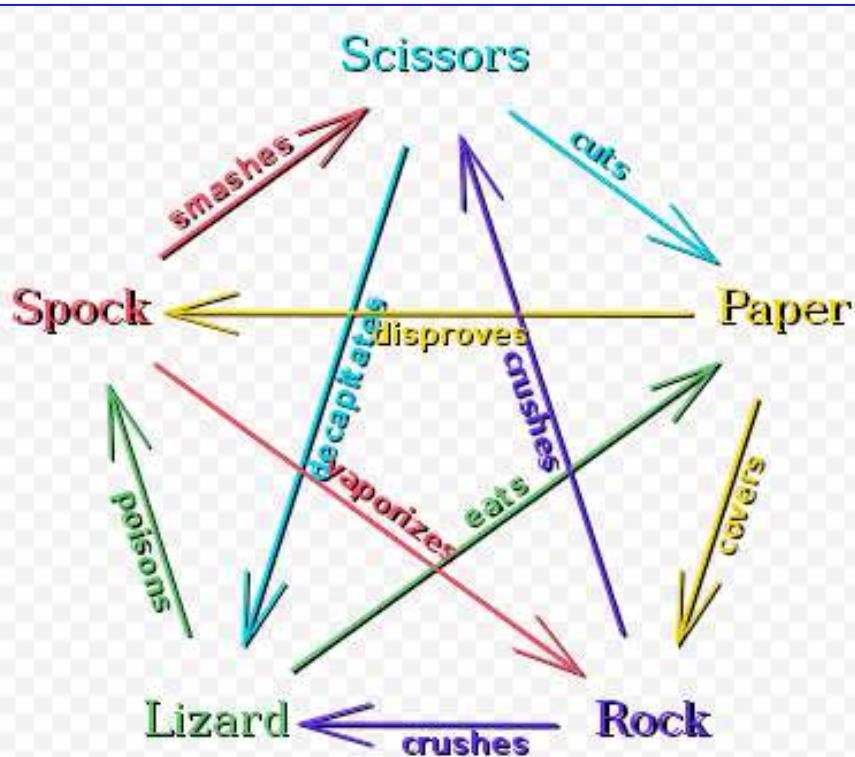


CS is just programming, right?



I'm not so sure...





rock – paper – scissors – lizard – Spock!

Let's play! Maybe
two out of three?



Logically, I've
got game!

Soundbite Syllabus

Lectures

T and Th: 8:10-9:25 am

Key dates: [Fall Break](#), [Midterm](#), [Final](#)
Instructor: [Professor](#) (Office hours: [MWF 10-11am](#))
Come to Lectures! (Why, how)
We'd like to see you! Let us know if you'll be sick...

Lab

recommended by 4 out of 5
CS5 alums!



T or W: 2:45 - 4:45pm or 6-8 pm

Guidelines: [Safety](#), [Equipment](#), [Protocols](#)
No lab without [lab pass](#) (Get it for lab)
With [lab pass](#), you have more time and effort in labs

Office hrs

F: 2:00-4:00 pm — Linda Activites Center Lab

Lots of help is available!

Or, come to any of the many tutoring ms.

HW

Monday Hw is due on Monday nights...

Syllabus, briefly

Lectures

T and Th: *8:10-9:25 am*

Key skills, topics, and their motivation

Insight into the HW problems (what, *why*, how)

We'd like to see you! Let me know if you'll be sick...

Lab

recommended by 4 out of 5
CS5 alums!



T or W: *2:45 - 4:45pm or 6-8 pm*

Guided progress on the week's hw

Not required, but encouraged: *full credit for lab*

Will **SAVE** you time and effort in CS 5

Office hrs

F: *2:30-4:30 pm, Linde Activities Center lab*

feel free to work on HW, to just stop by,
or, come to any of the **many** tutoring hrs!

HW

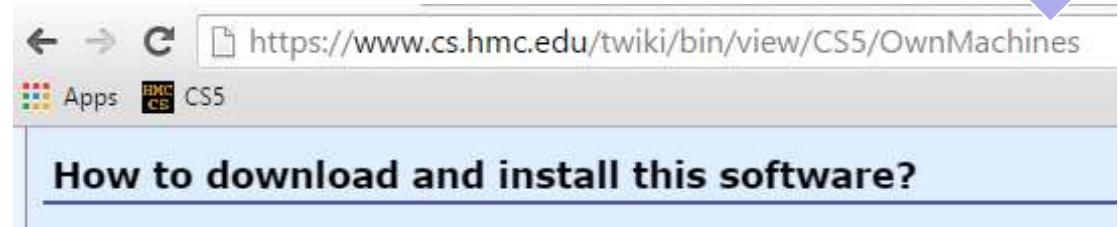
Monday nights: *due by 11:59 pm*

Each week's lab...

0) Find the lab! *Sign in...*

1) Get Python running...

download things
now, perhaps!



demo

2) Edit, run, + submit a file...

Encouraged: *bring your laptop*

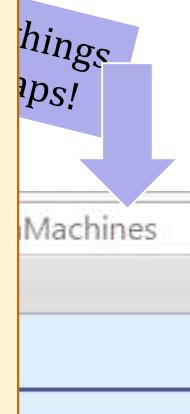
Each week's lab...

Labs are optional, but *incentivized*.

If you come to lab, give a good-faith effort, and sign in, you'll receive **full credit for the lab problems** even if you don't finish

den

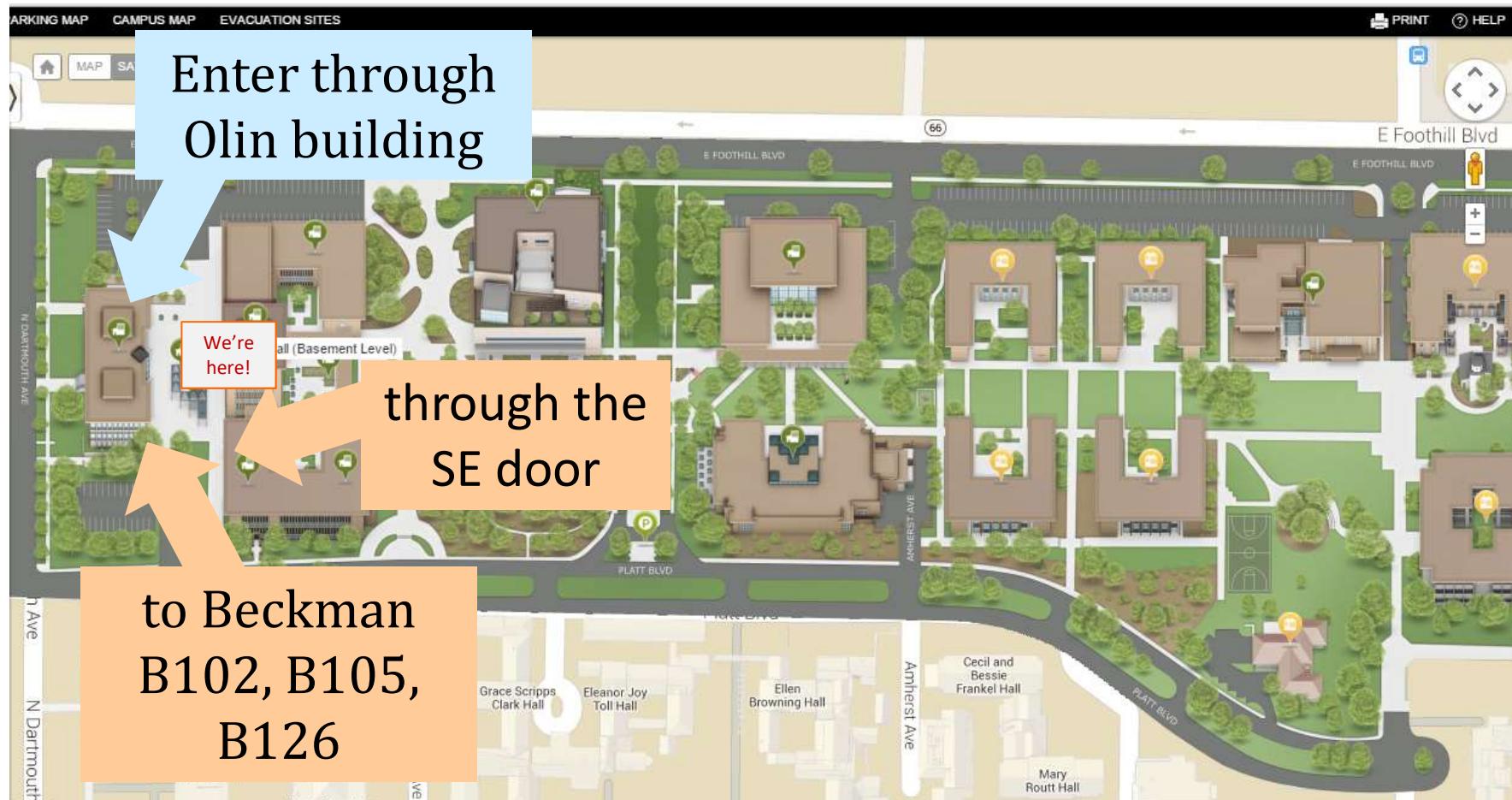
(you do need to submit by the usual hwk due date)



Encouraged: *bring your laptop*

Evening lab?

Olin's Southeast door is open!



Map to CS Lab

Shan

coffee

Laptop? **Bring it!**

Koi

Edwards

Macalister

Pryne

cool machines - drills, lathes, etc.

other keyboard-free machines

B102

B100

CS Hallway and Labs

Big
Beckman
(B126)

B105

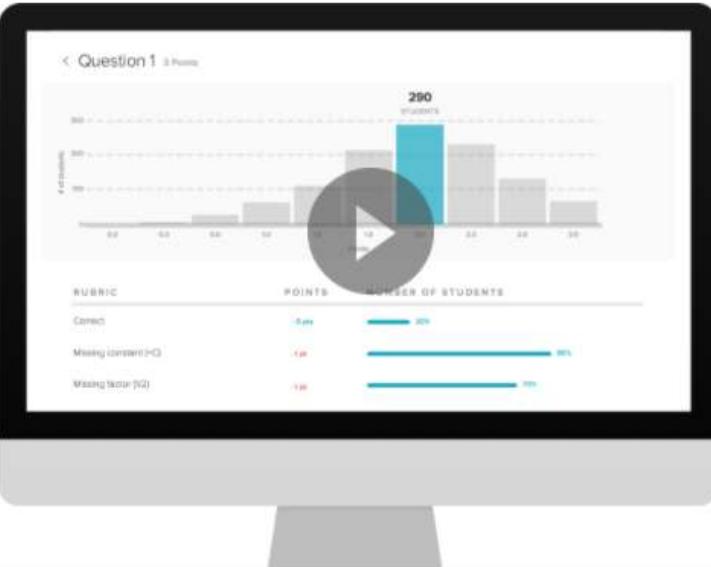
Galileo

Beckman

Biologists, bees,
spiders and other
arachnophiles

to Olin (Bio + CS)

Submissions: *GradeScope*



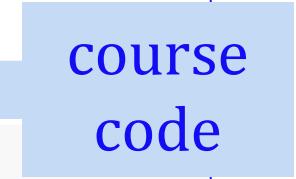
Sign up for free 

Grade Faster. Teach Better.

Gradescope streamlines the tedious parts of grading so you can focus on what you do best: giving great feedback.

Sign up for free 

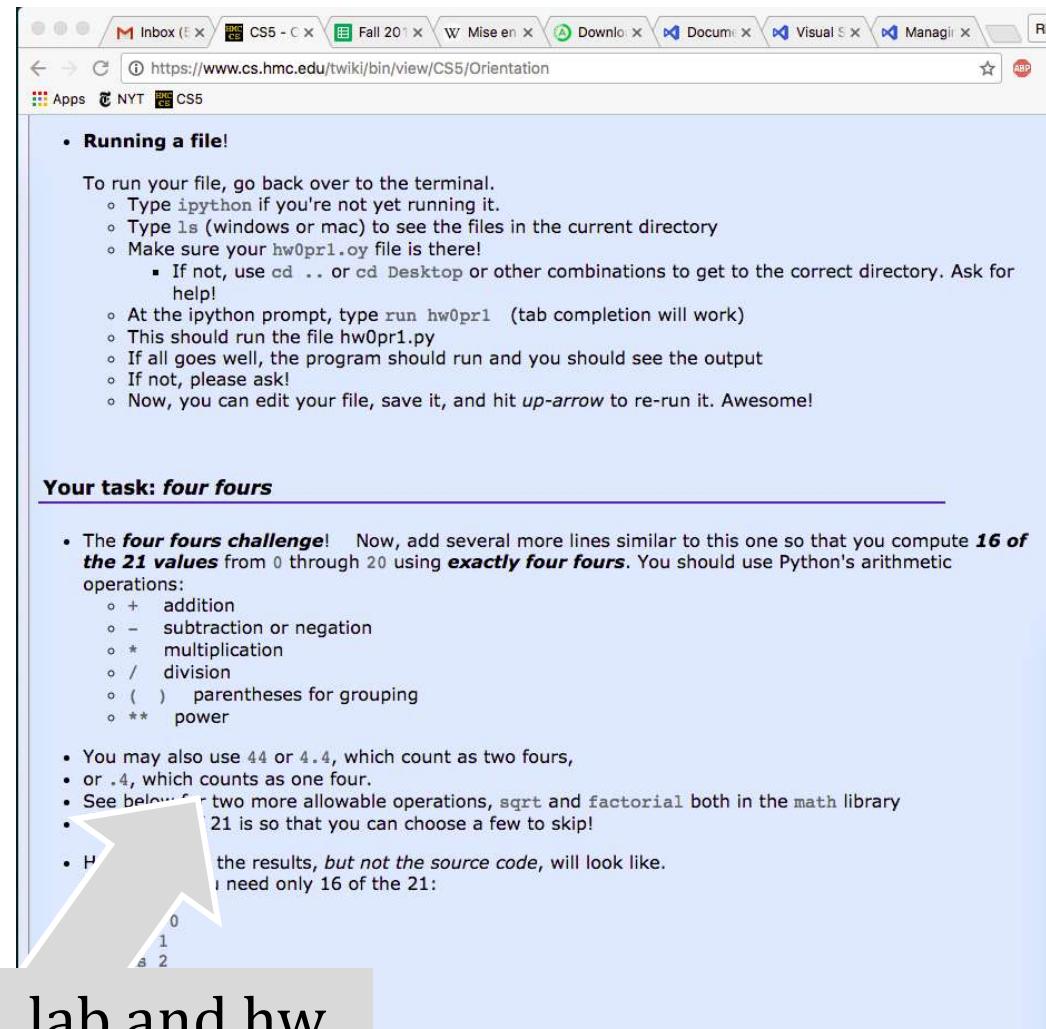
View Plans >

MPVY6M 

Happy Instructors at Over 300 Schools

This week: Lab 0



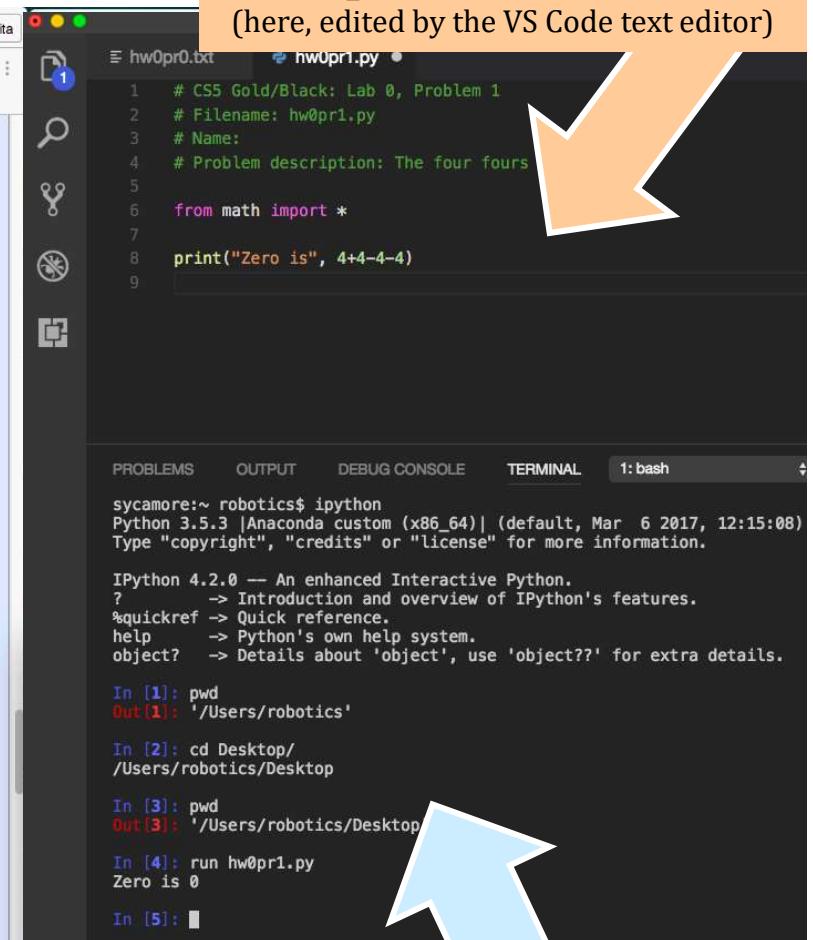
Your task: four fours

- The **four fours challenge!** Now, add several more lines similar to this one so that you compute **16 of the 21 values** from 0 through 20 using **exactly four fours**. You should use Python's arithmetic operations:
 - + addition
 - subtraction or negation
 - * multiplication
 - / division
 - () parentheses for grouping
 - ** power
- You may also use .4 or 4.4, which count as two fours,
- or .4, which counts as one four.
- See below for two more allowable operations, sqrt and factorial both in the math library
- There are 21 so that you can choose a few to skip!
- Hint: the results, but not the source code, will look like.
I need only 16 of the 21:

lab and hw
instructions

Python source code,
a plain-text file

(here, edited by the VS Code text editor)



```
# CS5 Gold/Black: Lab 0, Problem 1
# Filename: hw0pr1.py
# Name:
# Problem description: The four fours
from math import *
print("Zero is", 4+4-4-4)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash

```
sycamore:~ robotics$ ipython
Python 3.5.3 |Anaconda custom (x86_64)| (default, Mar  6 2017, 12:15:08)
Type "copyright", "credits" or "license" for more information.

IPython 4.2.0 -- An enhanced Interactive Python.
?           --> Introduction and overview of IPython's features.
%quickref --> Quick reference.
help       --> Python's own help system.
object?   --> Details about 'object', use 'object??' for extra details.

In [1]: pwd
Out[1]: '/Users/robotics'

In [2]: cd Desktop/
Out[2]: '/Users/robotics/Desktop'

In [3]: pwd
Out[3]: '/Users/robotics/Desktop'

In [4]: run hw0pr1.py
Zero is 0

In [5]:
```

shell or command-
line or terminal
(the execution environment)

get everything running *on your own machine*

Homework

Assignments

~ 5 problems/week

Due **Monday** evenings by 11:59 pm.

Extra credit is usually available...

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

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



No need to let us know, even.

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!)
- If you do, you must share the work equally - typing and coaching
- Be sure to indicate who your partner was at the submission site!

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*

Submit with a partner as ***full co-owners*** of the work.

Honor Code

- 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 duo.

Even with
three eyes, I
need to
borrow
others' to find
the syntax
errors here!



Sign & submit CS's honesty policy **online** in this week's lab.

Grading

~ 65% Assignments

~ 30% Exams

~ 5% Participation/“quizzes”

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

many take
cs5 P/NC

see online syllabus for the full grade list...

Exams

Midterm

Final

Th, Nov. 2, in-class

W or Th, Dec 13th (7pm) or 14th (2pm)

Midterm? This feels
more like a 2/3-term!



using a page of notes
is OK on exams

the exams are *written*,
not coded

the problems are modeled on
the in-class "quizzes"

Choices, choices!

Let's set the value of **perc** to 0.91...

↓
perc = 0.91

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

What will this program print,
if **perc** is 0.91?

First – who sees the
syntax errors here !?

What's here?

here:

here:

here:

Choices, choices!

Let's set the value of **perc** to 0.91...

↓
perc = 0.91

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

What will this program print,
if **perc** is 0.91?

Lots of Illuminating Solid
Parentheses!



Aargh! ;-)

What's here?

of **BLOCKS** here:

of **TESTS** here:

of **CONTROL STRUCTURES** here:

Choices, choices!

Let's set the value of **perc** to 0.91...

↓
perc = 0.91

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

What will this program print,
if **perc** is 0.91?

What's here?

of BLOCKS here:

of TESTS here:

of CONTROL STRUCTURES here:

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 `perc` is 0.8?

What value of `perc` gives an '`A-`' on the right?

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

Exclusive Choices

if ... elif ... else

```
if perc > 0.95:  
    print('A')  
  
elif perc > 0.90:  
    print('A-')  
  
elif perc > 0.70:  
    print('Pass')  
  
else:  
    print('Aargh!')  
  
elif and else are optional
```



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

Exclusive Choices

```
if p < 0.5:  
    print('A')  
  
elif p > 0.90:  
    print('B')  
  
elif p > 0.70:  
    print('C')  
  
else:  
    print('D')  
  
elif and else are optional
```

*Every **if** starts a new control structure.*

4 mutually exclusive blocks

*Every **elif** and **else** continues an existing control structure.*

*the first whose test is **True**.
If all fail, the **else** will run*

What's the difference?

mutually exclusive blocks

perc

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

nonexclusive blocks

perc

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

What if **perc == .99** ? (How would we set it?)

How many separate **control structures** does each side have?

What's the difference?

mutually exclusive blocks

perc

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

1

thing

*non*exclusive blocks

perc

```
→ if perc > .95:  
    print('A')  
  
→ if perc > .90:  
    print('A-')  
  
→ if perc > .70:  
    print('Pass')
```

3

things

What if `perc == .99` ? (How would we set it?)

How many separate **control structures** does each side have?

Nesting for decision-making, we now have it ***all***...



Nesting for decision-making, we now have it ***all***...



Nesting

for *decision-making*, we now have it ***all***...



So, let's catch 'em ***all***...

Nesting

Does this program print the correct RPS result this time?
Does it always?

```
comp = 'rock'  
user = 'paper'
```

```
if comp == 'paper' and user == 'paper':  
    print('We tie. Try again?')
```

```
elif comp == 'rock':  
  
    if user == 'scissors':  
        print('I win! *_*')  
    else:  
        print('You win. Aargh! ') # Blocks ?
```

Tests ?

C. Structures ?

Pair up with someone nearby – answer these questions together...

"Quiz"

Name _____

Name _____



Your favorite _____ is _____.

Your favorite _____ is _____.

Your least favorite _____ is _____.

Your least favorite _____ is _____.

What is something non-Claremont-collegy you have in common?

Then, try these Python q's:

(0) Find the 3 tests and 4 blocks here.

(1) What does this code print?

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
    if user == 'paper':  
        print('I win *_*!')  
    elif user == 'scissors':  
        print('You win.')  
else:  
    print('Tie.')
```

(2) As written, what output does this print?

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
    print('I win *_*!')  
if user == 'paper':  
    print('You win.')  
else:  
    print('Tie.')
```

(3) **Change** these inputs to produce a completely correct RPS output here.

(4) How many of the 9 RPS **input cases** are *fully correctly* handled here?

comp	'rock'	'paper'	'scissors'
user			
'scissors'			
'paper'			

(5) What is the **smallest** number of **blocks** and **tests** you'd need for a full game of RPS?

(Extra) What if it were RPS-5, which includes Lizard and Spock? How about RPS-101?

Pair up with someone nearby – answer these questions together...

Name _____

Your favorite _____

Your least favorite _____

People

What is something non-Claremont-collegy you have in common?

Then, try these Python q's:

(0) Find the 3 tests and 4 blocks here.

(1) What does this code print?

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
    if user == 'rock':  
        print('I win.')  
    elif user == 'scissors':  
        print('You win.')  
    else:  
        print('Tie.')
```

Python

(2) As written, what output does this print?

```
comp = 'rock'  
user = 'rock'
```

```
'rock':  
    win *_!*'  
paper':  
    'You win.'  
else:  
    print('Tie.')
```

(5) What is the *smallest* number of **blocks** and **tests** you'd need for a full game of RPS?

(Extra) What if it were RPS-5, which includes Lizard and Spock? How about RPS-101?

Name _____

Your favorite _____ is _____.

Your least favorite _____

Paper



"Quiz"

(3) **Change** these inputs to produce a completely correct RPS output here.

(4) How many of the 9 RPS **input cases** are *fully correctly* handled here?

comp	'rock'	'paper'	'scissors'
user			
'scissors'			
'paper'			
'rock'			

Pair up with someone nearby – answer these questions together...

"Quiz"

Name _____

Name _____



Your favorite _____ is _____.

Your favorite _____ is _____.

Your least favorite _____ is _____.

Your least favorite _____ is _____.

What is something non-Claremont-collegy you have in common?

Then, try these Python q's:

(0) Find the 3 tests and 4 blocks here.

(1) What does this code print?

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
    if user == 'paper':  
        print('I win *_*!')  
    elif user == 'scissors':  
        print('You win.')  
else:  
    print('Tie.')
```

(2) As written, what output does this print?

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
    print('I win *_*!')  
if user == 'paper':  
    print('You win.')  
else:  
    print('Tie.')
```

(3) **Change** these inputs to produce a completely correct RPS output here.

(4) How many of the 9 RPS **input cases** are *fully correctly* handled here?

comp	'rock'	'paper'	'scissors'
user			
'scissors'			
'paper'			

(5) What is the **smallest** number of **blocks** and **tests** you'd need for a full game of RPS?

(Extra) What if it were RPS-5, which includes Lizard and Spock? How about RPS-101?

"Quiz"

- Name Zach Dodds
- Your favorite ^{tv} show is Modern Family

+ Dr.
Who



- Your least favorite coffee is decaffeinated

- Name T. E. Alien

- Your favorite canned-meat food product is spam

- Your least favorite # is 41.999
so close!



Something in common?

Our taste in hats!

Pair up with someone nearby – answer these questions together.

Name _____

Your favorite _____

You _____

What _____

The _____

(0) Find the _____

(1) What does _____

```
comp = 'rock'  
user = 'paper'
```

```
if comp == user:  
    if user == 'rock':  
        print('Tie')  
    elif user == 'paper':  
        print('User wins')  
    else:  
        print('Computer wins')
```

What is the **smallest** number of **blocks** and **tests** you'd need for a full game of RPS?

(Extra) What if it were RPS-5, which includes Lizard and Spock? How about RPS-101?

"Quiz"



Please pass these up
the aisles...

(take a picture, if you'd like)

... then, turn back to
the notes

sors	scissors	paper

"Quiz" ~ problems 1+2

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
  
    if user == 'paper':  
        print('I win *_*!')  
    elif user == 'scissors':  
        print('You win.')  
  
else:  
    print('Tie.')  
    print('Ties go to the runner.')  
    print(' - and I am running!')
```

... what if this **else** block were indented?

"Quiz" ~ problems 3-5

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
    print('I win *_*!')  
  
if user == 'paper':  
    print('You win.')  
  
else:  
    print('An awful tie')
```

What does this program print?

"Quiz" ~ problems 3-5

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
    print('I win *_*!')  
  
if user == 'paper':  
    print('You win.')  
  
else:  
    print('An awful tie')
```

		comp
		'rock' 'paper' 'scissors'
user	'rock'	
	'paper'	
	'scissors'	

How many possible “input cases” are there?

For how many is this program correct?

How **efficient** can we be?
For RPS-3? RPS-5? RPS-101?

"Quiz" ~ problems 3-5

```
comp = 'rock'  
user = 'rock'  
  
if comp == 'rock':  
    print 'I win *_*'  
  
if user == 'rock':  
    print 'Tie'  
else:  
    print 'You win!'
```

user	comp
'rock'	'rock'
'paper'	'paper'
'scissors'	'scissors'

How many possible “input cases” are there?

For how many is this program correct?

How **efficient** can we be?

For RPS-3? RPS-5? RPS-101?

Remember ~ Lab this week

Tue. or Wed. ~ afternoon or evening

Bring your laptop to Beckman B126 (here)

- or use one of the CS machines in B105

Get started with Python/text editor/cmdline...

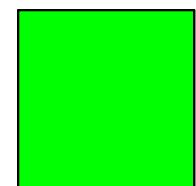
See you in lab!

(perhaps at 2:44:44 today...?)

though it's more than a few bits early!



**Alien defeats everything –
*even Alien***



How about a sneak peek at this week's HW... ?

... you must mean sneak *Pic* !