Welcome to CS 5!

xkcd, CS's id
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

We give this course two thumbs...

- Metametacritic

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

1 handout...

slides & syllabus
Welcome to CS 5!

"Grab" these lecture notes...

Introduction to CS

Wally Wart, a protrusive advocate of concrete computing

Everyone will get out of this course – a lot!

We don't have words strong enough to describe this class.

- NYTimes Review of Courses
- US News and Course Report
- Metacritic
- MetaMetacritic

Today's Zoom-Chat Q'n...

What is your current city, country ... and timezone: What's your time right now?
Take-home message...

Yay! in 2020: just Google for hmc cs5

www.cs.hmc.edu/cs5
Break-out ice-breaker...

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

Name ____________________
Your favorite _________ is ___________.
Your least favorite ___________ is ___________.

Name ____________________
Your favorite _________ is ___________.
Your least favorite ___________ is ___________.

"Quiz"

Later on today...

Give me a break!

What is something non-Claremont-collegel 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?

```python
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?

```python
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?

(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?
Break-out ice-breaker...

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

Name ___________________________ Name ___________________________

Your favorite ________ is ___________. Your favorite ________ is ___________.

Your least favorite __________ is ___________. Your least favorite __________ is ___________.

What is something non-Claremont-college-y you have in common?

To do now...

"Quiz"

Then, try these Python q's:

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

(1) What does this code print?

```python
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?

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

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

(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?

(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?
Break-out ice-breaker...

Agree on common "attributes"

Later on today...
For an incorrect example:

Try out the Zoom-editing toolset.

Three people? No worries – add more!

Then, take a screenshot ...

+ Mac: Command-Shift-4
+ Win: built-in "Snipping Tool"

... and upload this to your GradeScope!
For an *incorrect* example:

Try out the Zoom-editing toolset.

Three people? No worries – add more!

Then, take a screenshot ... 

+ Mac: Command-Shift-4
+ Win: built-in "Snipping Tool"

... and upload this to your GradeScope!
Break-out ice-breaker...

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

Name _______________________
Your favorite ________ is __________.
Your least favorite __________ is __________.

Name _______________________
Your favorite ________ is __________.
Your least favorite __________ is __________.

"Quiz"

What is something non-Claremont-collegey 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('You lose...')
    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: Ugh')

(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?

(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?
Introductions...

Zach Dodds
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 ...

Robots

Outreach

Chairs?

Who?!?? Dinos!
You're here ~ what's next?

1) How CS 5 runs...

2) Python?!

the first Python HW is *choice!*

Shouldn’t there be an alien in this game?

3) What *is* CS?

CS is just programming, right?

I'm not so sure...

Whatever it is, it's definitely *alien*!
A minute of cs5 programming...

Lab 0: getting everything running on your own machine

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

lab and hw instructions

shell or command-line or terminal (the execution environment)
hw0pr1: *What's in Lab today?*

Might start on the sw download/install now...

Help! I'm stuck in this webpage – with spam!
A minute of cs5 programming...

Lab 0: getting everything running on your own machine
A minute of cs5 programming...

Lab 0: getting everything running on your own machine

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

Shell or command-line or terminal
(the execution environment)
A minute of cs5 programming...

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

Try it! (now?!)
A minute of cs5 programming...

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

Python source code, a plain-text file
(here, edited by the VS Code text editor)
Logically, I've got game!

Rock – paper – scissors – lizard – Spock!

Let's play! Maybe two out of three?

www.youtube.com/watch?v=fqlDc2VICZ0 start at about :28

hw0: rock-paper-scissors
Syllabus, briefly

Lectures

**MW:** 1:00-2:45pm

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"

**MW:** 3:00-5:00pm

Guided progress on the week's hw

We have Kevin, Malia, Deanna, and me...

these are recommended by 5 out of 5 CS5 alums!

Office and grutoring hrs

**Office hrs:** 7:00-9:00pm pacific

Deanna: 10:00am-noon pacific

Hw is due evenings...

Rapid!

Lots of help is available!
Syllabus, briefly

Lectures

**MW: 1:00-2:45pm**
- 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"

**MW: 3:00-5:00pm**
- Guided progress on the week's hw
- We have Kevin, Malia, Deanna, and me...

Office and grutoring hrs

**Office hrs:** 7:00-9:00pm pacific
- **Malia:** 10:00am-noon pacific
- **Deanna:** 10:00am-noon pacific

**hwk or other q'ns**

HW

**Rapid!** *due by 11:00 pm, soon after it's assigned!**

these are recommended by 5 out of 5 CS5 alums!
Submissions: GradeScope

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  Watch our video

MKP2E3  course code

Happy Instructors at Over 300 Schools
Homework

Assignments ~ 5 problems/day

Due various evenings by 11:00 pm.

Extra credit is available (fun!)

You have 100 CS 5 Euros to use...
"Uncounted points"

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.

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

~ 50% Assignments
~ 45% Exams
~ 5% Participation/“quizzes”

the exams are written, not coded (???)
the problems are modeled on the in-class "quizzes"

Exams

Midterm Fri, May 29, "in-class"
Final Fri, June 5, "in-class"

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

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...

using a page of notes is OK on exams
Choices, choices!

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

\[
\text{perc} = 0.91
\]

```python
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 – do you see the **syntax errors** here !?
Let's set the value of `perc` to 0.91...

```
perc = 0.91
```

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

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

What's here?

Aargh! ;-)
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:

how many tests are executed?
Choices, choices!

```python
perc = 0.80

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

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

```python
if perc > 0.95:
    print('A')

elif perc > 0.90:
    print('A-')

elif perc > 0.70:
    print('Pass')

else:
    print('Aargh!')
```

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

4 mutually exclusive blocks in a single control structure

`elif` and `else` are optional
Exclusive Choices

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

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

at most one block will run: the first whose test is **True**. If all fail, the **else** will run

4 mutually exclusive blocks

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

**elif** and **else** are optional
What's the difference?

<table>
<thead>
<tr>
<th>mutually exclusive blocks</th>
<th>nonexclusive blocks</th>
</tr>
</thead>
</table>

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

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

```python
perc

if perc > .95:
    print('A')

elif perc > .90:
    print('A-')

elif perc > .70:
    print('Pass')

if perc > .95:
    print('A')

if perc > .90:
    print('A-')

if perc > .70:
    print('Pass')
```
What's the difference?

mutually exclusive blocks

What if \texttt{perc} == .99? (How would we set it?)

How many separate \textit{control structures} does each side have?

\begin{align*}
\text{perc} &= .99 \\
\text{if} & \quad \text{perc} > .95: \\
& \quad \quad \text{print('A')} \\
\text{elif} & \quad \text{perc} > .90: \\
& \quad \quad \text{print('A-')} \\
\text{elif} & \quad \text{perc} > .70: \\
& \quad \quad \text{print('Pass')} \\
\end{align*}

\begin{align*}
\text{perc} &= .99 \\
\text{if} & \quad \text{perc} > .95: \\
& \quad \quad \text{print('A')} \\
\text{if} & \quad \text{perc} > .90: \\
& \quad \quad \text{print('A-')} \\
\text{if} & \quad \text{perc} > .70: \\
& \quad \quad \text{print('Pass')} \\
\end{align*}
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...
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!')
Pair up with someone nearby – answer these questions together...

Name ______________________ 
Your favorite __________ is ____________.
Your least favorite ____________ is ____________.

Name ______________________ 
Your favorite __________ is ____________.
Your least favorite ____________ is ____________.

What is something non-Claremont-collegey 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?

```python
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?

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

(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?

(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 __________ is ____________.

Your least favorite ____________ is ____________.

Name ______________________

Your favorite __________ is ____________.

Your least favorite ____________ is ____________.

What is something non-Claremont-collegey 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?

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

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

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

(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?

(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”

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

Name ______________________  Name ______________________

Your favorite tv show is ____________  Your favorite canned-meat food product is ____________.

Your least favorite coffee is ____________  Your least favorite # is ____________.

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

Our taste in hats!

Then, try these Python q’s:

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

(1) What does this code print?

```python
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?

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

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

(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?

(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 __________ is ____________.
Your least favorite ____________ is ____________.

Name ______________________
Your favorite __________ is ____________.
Your least favorite ____________ is ____________.

What is something non-Claremont-collegey 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?

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

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

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

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

if comp == 'rock':
    if user == 'paper':
        print('I win *_*!')
    elif user == 'scissors':
        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?

(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?
`comp = 'rock'
user = 'rock'

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

... what if this `else` block were indented?
comp = 'rock'
user = 'rock'

if comp == 'rock':
    print('I win *__*!')

if user == 'paper':
    print('You win.')

else:
    print('Tie: Ugh')

What does this program print?
```python
comp = 'rock'
user = 'rock'

if comp == 'rock':
    print('I win *_*!')

if user == 'paper':
    print('You win.')

else:
    print('Tie: Ugh')
```

"Quiz" ~ problems 3-5

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?
comp = 'rock'
user = 'rock'

if comp == 'rock':
    print('I win *_*!')

if user == 'paper':
    print('You win. ')

else:
    print('Tie: Ugh')

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?
comp = 'rock'
user = 'rock'

if comp == 'rock':
    print('I win *_*!')

if user == 'paper':
    print('You win.

else:
    print('Tie: Ugh

A correct RPS is possible with only

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

Name ______________________

Your favorite __________ is ____________.

Your least favorite ____________ is ____________.

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

Name ______________________

Your favorite __________ is ____________.

Your least favorite ____________ is ____________.

In Claremont... we'd "pass these up the aisles" (take a picture, if you'd like)

... now, we'll hand them into GradeScope ...
CS !\neq \text{ programming}
CS $\neq$ programming

"not equal to"
CS != programming

So, what is CS?
programming ≠ CS

programming : CS ::

longboards : HMC maybe 5Cs?
capital : business venture
equations : mathematics
language : ideas
web search : knowledge

Tesla : Google

programs are a **vehicle**, but not the destination
Today in CS5

1) How CS 5 runs...

2) Python?!

3) What *is* CS?

CS is just programming, right?

I'm not so sure...

Whatever it is, it's definitely *alien*!
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 *construct* in CS 5...
...at least to some extent!

3 examples? That's *it* for me!
What is CS?

Can you solve the problem?
How can it be done?
How well can it be done?
Can it be done at all?

Can you create a process to solve such problems?

What is the Longest Common Subsequence between 2 strings?

biology's string-matching problem, "LCS"

'HUMAN'
'CHIMPANZEE'

'CGCTGAGCTAGGCC...'
'ATCCTAGGTAACTG...'
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?

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

Only one is programming. Which one?

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**

Only one is programming.  *Which one?*

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!
Remember ~ Lab today

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

See you in lab!
(perhaps it's 2:44:44 today...?)

though this is more than a few bits early!

Alien defeats everything – even Alien

How about a peek at the rest of the week's HW...?

... you must mean Pic!