

ECS 189C: Software Correctness

Lecture 0 – WQ2026

Welcome!

- This is a **special topics course**
- Listed title: “Special topics in programming languages & compilers”
- See <https://cs.ucdavis.edu/course-special-topics>
- Cap: 65 BUT some will drop – more expected off waitlist
- 4 units



Please remember to change the number of units to 4

Instructor

Your Instructor: Prof. Stanford

Started at Davis: July 2023



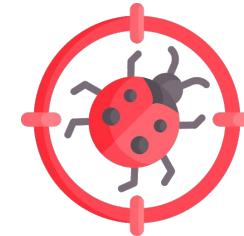
DavisPL Research Group

Plan for today

1. What is this class about?
2. Short discussion/activity
3. FAQ, syllabus, and logistics
4. Demos (time permitting)

What is this class about?

What is this class about?



We live in a world of **buggy**
software

Crashes... Crowdstrike (2024)

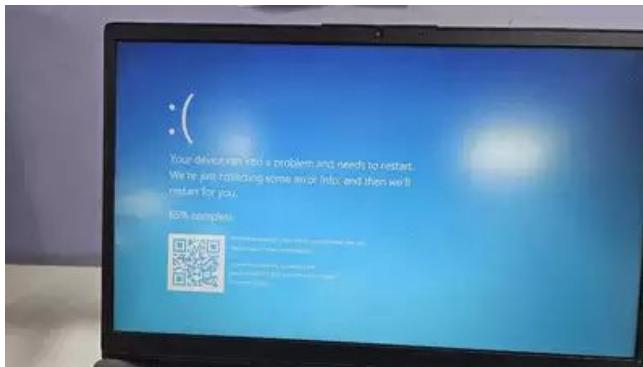
CYBERSECURITY

\$1.94B in Expected Healthcare Losses Due to CrowdStrike Disruption

CrowdStrike released a root cause analysis of the incident that caused a global outage on July 19

Pietje Kobus

Aug. 13, 2024

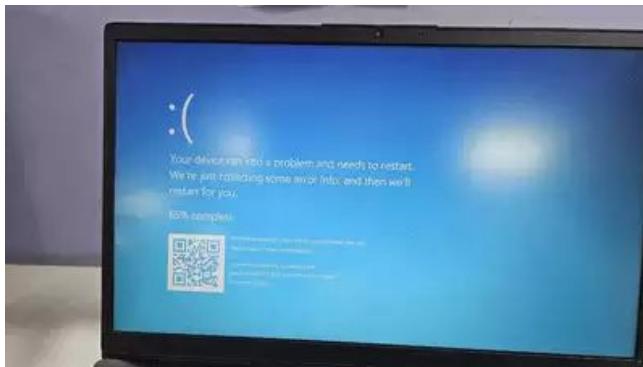


Crashes... Crowdstrike (2024)

CYBERSECURITY

\$1.94B in Expected Healthcare Losses Due to CrowdStrike Disruption

8.5 million devices were confirmed affected by the CrowdStrike outage, but Microsoft says that's only a subset.



Crashes... Crowdstrike (2024)

Crowdstrike → device driver → operated in kernel mode (full admin access)

- Crowdstrike driver certified by Windows (WHQL) as being compatible!
- However, driver was written to **load definition files** at runtime – “Agile development” “customers should get the latest protection” – see: Zero day attacks
- Invalid definition file (just contained 0s) - accessed an invalid pointer (0x9c)
 - Definitions can execute arbitrary code and no longer certified!
- Driver marked as a “**boot start driver**” – couldn’t be turned off!

Full explanation here: <https://www.youtube.com/watch?v=wAzEJxOo1ts>

Crashes... (September 2025)



NBC BAY AREA

BART

Computer problem halts all BART trains

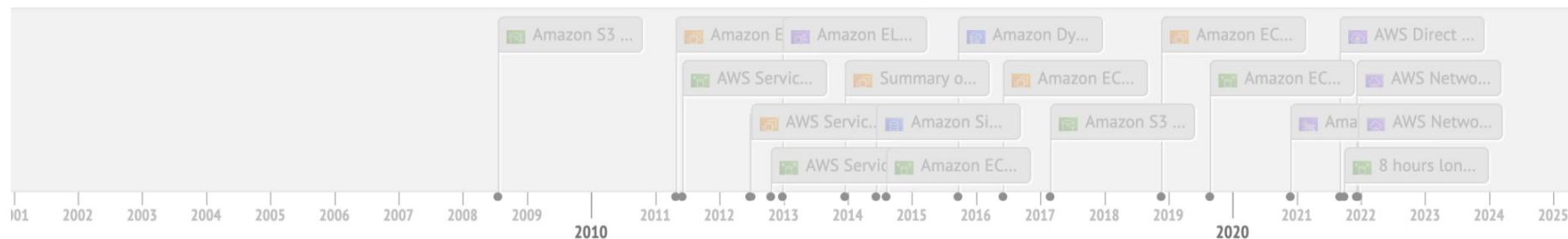
By Bay City News • Published 2 hours ago • Updated 14 mins ago



NBC Universal, Inc. BART trains were unable to start running early Friday morning after a computer equipment problem shut down the system. Bob Redell, Ginger Conejero Saab and Cinthia Pimentel report.

Outages... AWS (2010 to present)

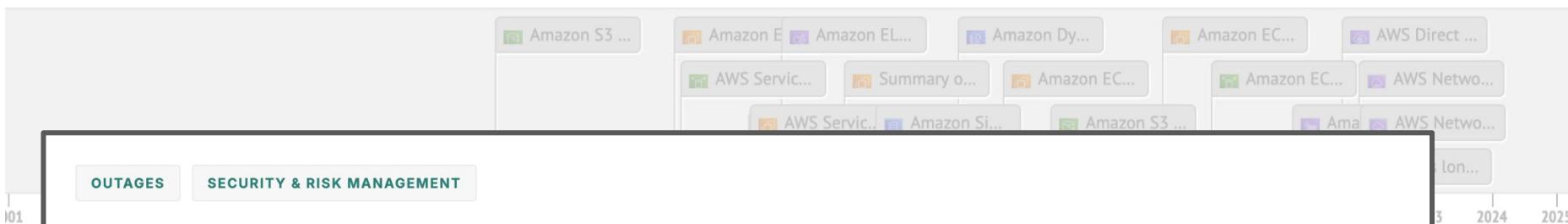
Timeline



Source: <https://awsmaniac.com/aws-outages/>

Outages... AWS (2010 to present)

Timeline



AWS Outage that Broke the Internet Caused by Mistyped Command

Amazon says Tuesday's mayhem resulted from mistake during a routine debugging exercise.

Source: <https://awsmaniac.com/aws-outages/>

Lost data... MacOS Sonoma

hoakley / March 18, 2024 / [Macs](#), [Technology](#)

Serious bug in Sonoma 14.4 will destroy saved versions in iCloud Drive

Lost data... Google AI (December 2025)



r/technology • 5h ago
lurker_bee

...

Google's Agentic AI wipes user's entire HDD without permission in catastrophic failure — cache wipe turns into mass deletion event as agent apologizes: "I am absolutely devastated to hear this. I cannot express how sorry I am"

Artificial Intelligence

Security bugs... Heartbleed

The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users and to impersonate services and users.

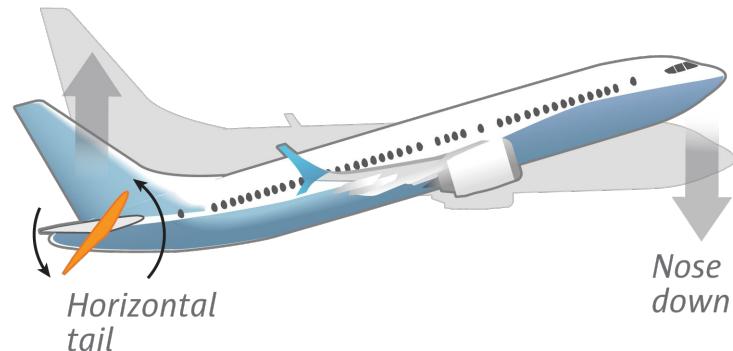


And even: Loss of life

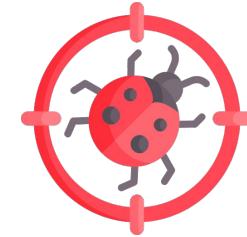


Therac-25
(1985-1987, 6 deaths)

MCAS safety system engages



Boeing 737 MAX
(2018-2019, 345 deaths)



We live in a world of **buggy** software

You might have even encountered your own bugs “in the wild”...

File syncing service (Oct 2018)

On Mon, Oct 22, 2018 at 02:33 AM, "Caleb Stanford" <caleb_stanford@alumni.brown.edu> wrote:

...

Hello,

I would like to report a bug in which Insync deleted 2 of my files while running solely in the background. I have included information below. After a progressive panic attack trying to find the files, I was finally able to find them in the .insync-trash folder after looking at this page <https://help.insynchq.com/resolving-and-reporting-issues/general/retrieving-deleted-or-missing-files>.

File syncing service (Oct 2018)

On Mon, Oct 22, 2018 at 02:33 AM, "Caleb Stanford" <caleb_stanford@alumni.brown.edu> wrote:

...

Hello.

Steps to reproduce:

1. Start with folder A with many files and a folder B somewhere else
2. Move the files from folder A to folder B by a single cut-and-paste operation
3. While Insync is still processing this move, delete folder A
4. Insync may delete files from folder B, after they are already moved by the system, if the file no longer exists in folder A. The files will be moved to `.../.insync-trash/.../B` and can be recovered from there.

Social media (May 2021)



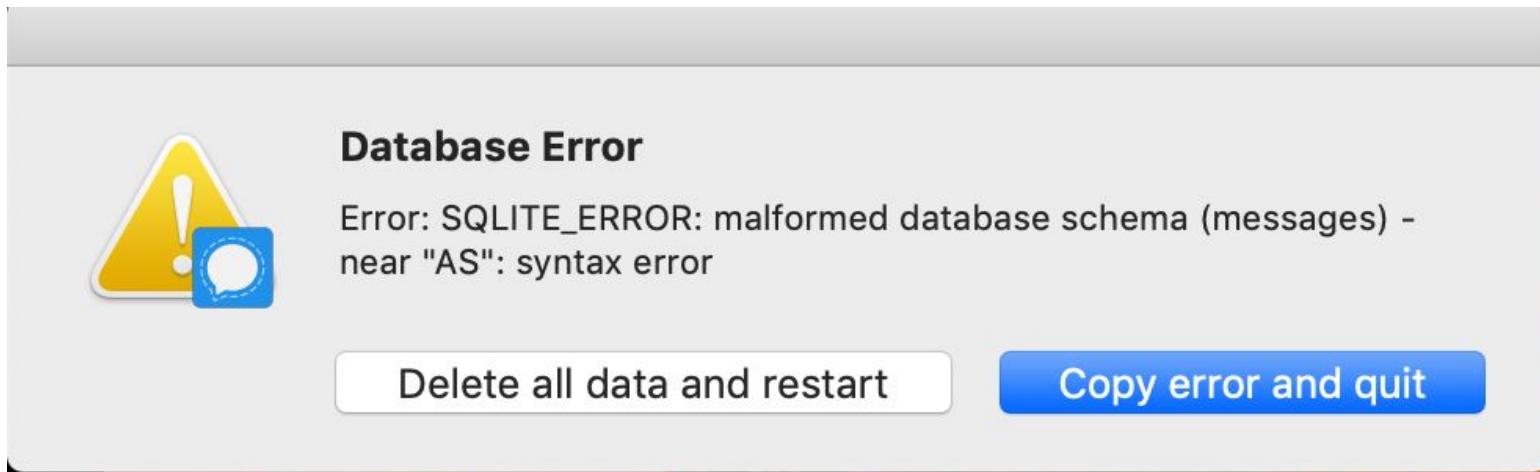
Activity Log

Today

Caleb Stanford liked [REDACTED] s comment. ...

Caleb Stanford liked [REDACTED] s comment. ...

Signal messaging app (March 2022)





(Unknown from January 2019)



The image shows a screenshot of an app's page on a digital store. At the top, there are four circular icons: a green one with '1 MILLION' downloads, a green hexagonal one with '4.1' stars, a green one with a wrench (Tools), and a teal one with a document (Similar). Below these are the labels 'Downloads', '22,110 ▲', 'Tools', and 'Similar'. A large green box contains the 'WHAT'S NEW' section, which lists: '- Updated third-party libraries', '- Many changes under the hood', '- Fixed bugs', and '- Added more bugs to fix later'. The last item is highlighted with a white border.

WHAT'S NEW

- Updated third-party libraries
- Many changes under the hood
- Fixed bugs
- Added more bugs to fix later

AI error (January 2025)



Yes, the **Travelpro Maxlite 5 Rolling Underseat Compact Carry-On** is specifically designed to meet budget airlines' personal item size restrictions. At 15.25" x 13" x 8", it fits within the typical personal item limits of:

- **Frontier Airlines:** 14" x 18" x 8" ✓



Omi

★★★★★ **Beautiful and compact**

Reviewed in the United States on September 18, 2023

Color: Orchid Pink Purple | **Verified Purchase**

Purchased this for my many trips for work. For me it held enough clothes for a week (I did laundry) and my 2 laptops. It swiveled easily thru the airport and for the most part all went well. I normally fly via **Frontier** and now they have gotten absurd with the drop your carryon into the preformed metal size thingee and this suitcase didn't fit. I told them I had 2 laptops which were unforgiving for squishing in down in size and they instructed me to remove them from the suitcase so I did and it fit comfortably in the size requirement. Then I was instructed that I could not replace them back into the suitcase but needed to "carry them aboard". Stupid... So I walked onto the ramp, opened my suitcase and put them back in. No issue with that. Then I got to my seat and found that with or without the laptops there was NO WAY this suitcase would fit UNDER the seat in front, so I put it top side. So if you're thinking it will fit under it will not but it fits easily above. The suitcase is very nice, I will continue to use it for my flights but now I know not to even try to put it under and to remove my laptops prior to boarding!

17 people found this helpful



Thank you for your feedback.

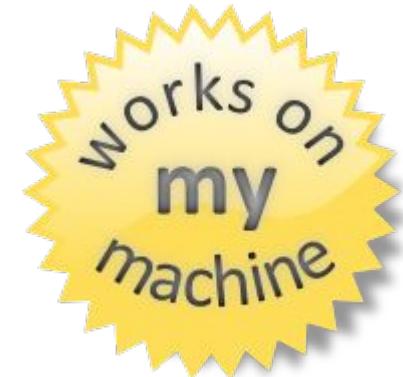
| Report



We live in a world of **buggy**
software

Making the situation worse...

The bug may only show up on **some platforms**

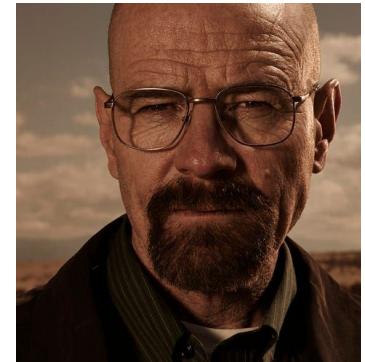


It may require an **esoteric/obscure** input



Or fail to show up at all.

Heisenbug (n.): a software bug that seems to disappear or alter its behavior when one attempts to study it.



Heisenbugs

Heisenbug (n.): a software bug that seems to disappear or alter its behavior when one attempts to study it.

▲ Ubuntu Bug 255161: Openoffice can't print on Tuesdays (launchpad.net)

244 points by franz e on Aug 13, 2014 | hide | past | favorite | 37 comments

Infinite loop heisenbug: it exits if I add a printout

Asked 9 years, 1 month ago Modified 3 years, 1 month ago Viewed 2k times

Game client (December 2025)

“Here’s a simple explanation of what’s happening: **the client is placing you into two games at once. You join the first game, and then moments later you’re placed into a second game.** You play the second game normally, but when the five-minute timer runs out in the first game, that opponent is prompted to make you resign. Once they click it, you’re pulled out of the second game and taken to the resignation screen for the first game.

If you start another game, you end up in a third game. At that point, you’re now in both the second and third games, and the same cycle repeats.”

(from a user on Reddit)

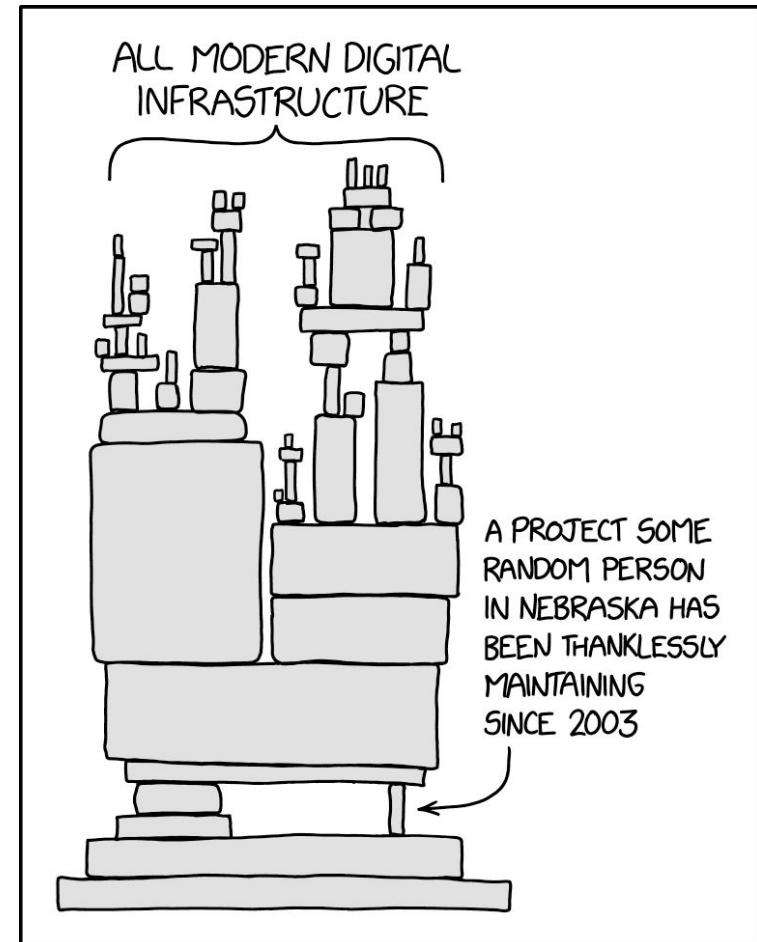
Why is software like this?



Why is software like this?



Why is software like this?



Credit: xkcd

Is there a better way?



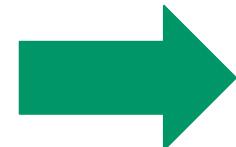
Is there a better way?



1. Write down what the program **does** (and should do)
2. Use **automatic tools** check (verify) whether the program meets its requirements

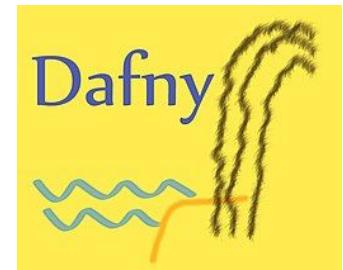
Is there a better way?

What this
class is
about



1. Write down what the program does (and should do)
2. Use automatic tools check (verify) whether the program meets its requirements

Real-world tools
used in industry



Discussion question

1. Describe what the program **does**
2. Come up with a definition for what this program **should do**
3. Share your definition with your neighbors



Then fill out this poll:

<https://forms.gle/TNVmBxNfeENx5nh79>

```
def is_even(x):  
    if x == 0:  
        return True  
    elif x == 1:  
        return False  
    elif x == 2:  
        return True  
    elif x == 3:  
        return False  
    elif x == 4:  
        return True  
    else:  
        return False
```

Sharing your answers

1. Describe what the program **does**
2. Come up with a definition for what this program **should do**
3. Share your definition with your neighbors



Then fill out this poll:

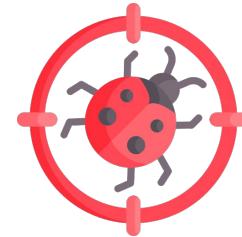
<https://forms.gle/TNVmBxNfeENx5nh79>

```
def is_even(x):  
    if x == 0:  
        return True  
    elif x == 1:  
        return False  
    elif x == 2:  
        return True  
    elif x == 3:  
        return False  
    elif x == 4:  
        return True  
    else:  
        return False
```

The underlying question: **What is a bug?**



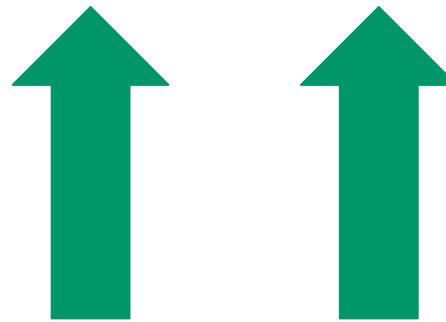
The underlying question: **What is a bug?**



Answer: It Depends!

(Remember that first step)

1. Write down what the program **does** (and should do)



Fundamental question in software development!
Difficult to answer precisely!

Example

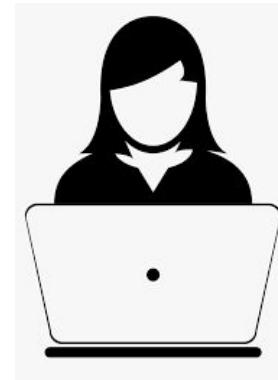


Please build me a car

Example



Please build me a car



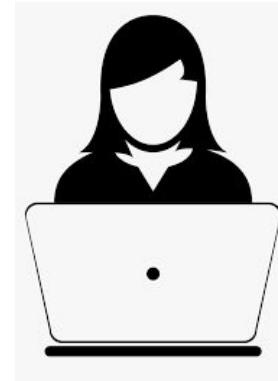
What car? What is the definition you have in mind?

Example



Please build me a car

It should have four
wheels and you can
drive it places



What car? What is the
definition you have in
mind?

Example



Please build me a car

It should have four
wheels and you can
drive it places



Ok, here you go

Example



Please build me a car

... it should also have a roof and seats inside



Ok, here you go

Example



Please build me a car

... it should also have a roof and seats inside



Ok, here you go

Example



Please build me a car

... it should also have a roof and seats inside

... etc.



Ok, here you go

Example 2



Please write a program
to check if a number is
even or odd

```
def is_even(x):
    if x == 0:
        return True
    elif x == 1:
        return False
    elif x == 2:
        return True
    elif x == 3:
        return False
    elif x == 4:
        return True
    else:
        return False
```

Ok, here you go

Example 2



Please write a program
to check if a number is
even or odd

```
def is_even(x):
    if x == 0:
        return True
    elif x == 1:
        return False
    elif x == 2:
        return True
    elif x == 3:
        return False
    elif x == 4:
        return True
    else:
        return False
```



Is this correct?

(Why not?)

Ok, here you go

Example 2



Please write a program
to <do some task>

A screenshot of a code editor showing a Python function definition:

```
def is_even(x):
    if x == 0:
        return True
    elif x == 1:
        return False
    elif x == 2:
        return True
    else:
        return False
```

The code is displayed on a black background with syntax highlighting. To the right of the code is a green circular icon containing a white neural network or knot-like symbol.

Ok, here you go



Is this correct?
(Why not?)

A screenshot of a conversational AI interface. The AI has apologized for a previous mistake and provided corrected code:

I apologize for the mistake in my previous response. You are correct that the output I provided was not correct. Thank you for bringing it to my attention.

Here is the corrected code to standardize the data:

```
python
```

[Copy code](#)

Like Dislike

Example 3

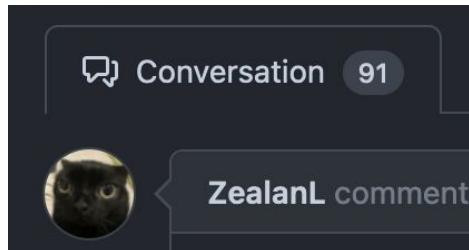


Please write a program
to play chess



Ok, here you go

Example 3



Your program can be used by a bad actor to access and modify arbitrary user memory?

vdbergh commented on May 6, 2023 · edited

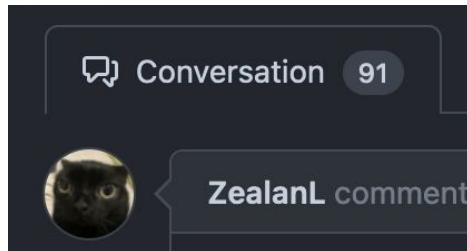
This is discussed in the FAQ.

<https://github.com/official-stockfish/Stockfish/wiki/Stockfish-FAQ#stockfish-crashed>

Summary. Stockfish is free to crash on any illegal position where "illegal" being defined as being not reachable from the starting position.

It's a feature, not a bug

Example 3



Your program can be used by a bad actor to access and modify arbitrary user memory?

vdbergh commented on May 6, 2023 · edited

This is discussed in the FAQ.

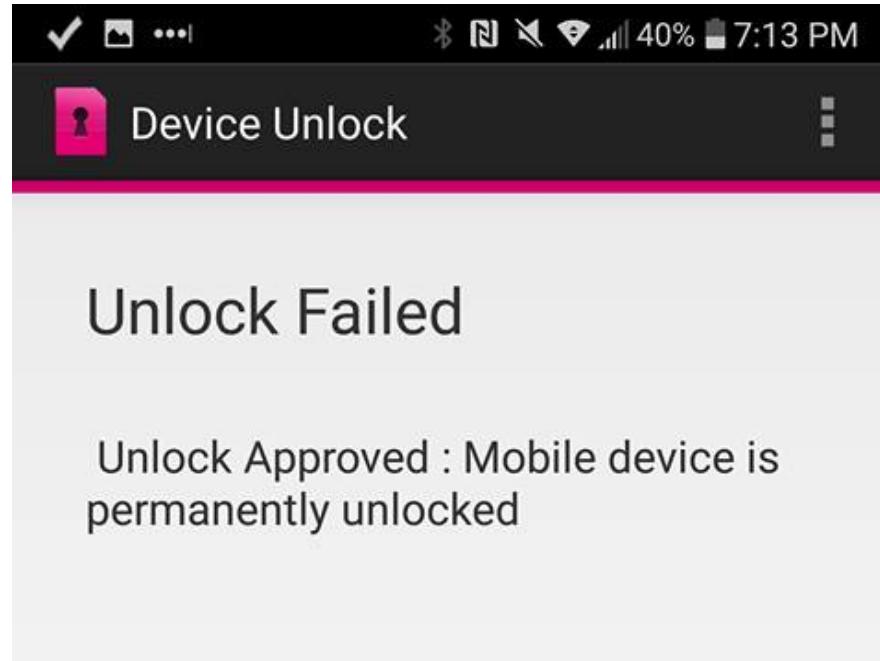
<https://github.com/Stockfish-Team/FAQ#stockfish>

Summary. Stockfish is being defined

Closed

It's a feature, not a bug

Example 4: is this a bug?



The problem

We need a

clear and unambiguous

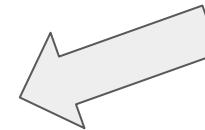
way to determine if programs are correct.

The problem

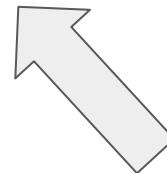
We need a

clear and unambiguous

way to determine if programs are correct.

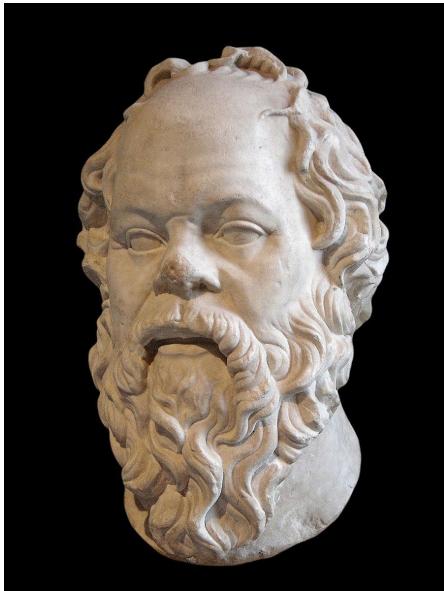


Everyone should
agree!



That is:
What the software is;
What it is supposed to do; and
Why it works (or why it doesn't)

Clear and unambiguous?



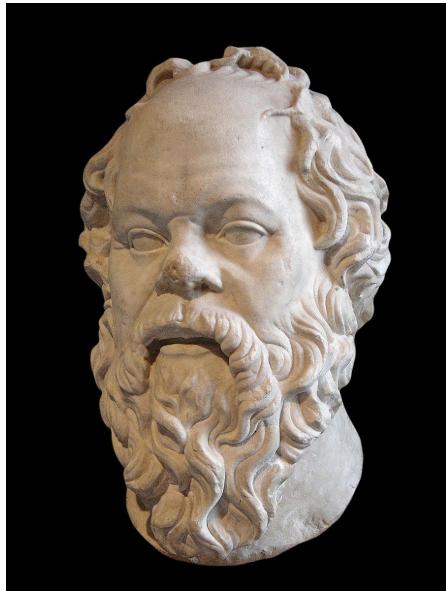
All men are mortal.

Socrates is a man.

Therefore, Socrates is mortal.^[2]

Logical Syllogism

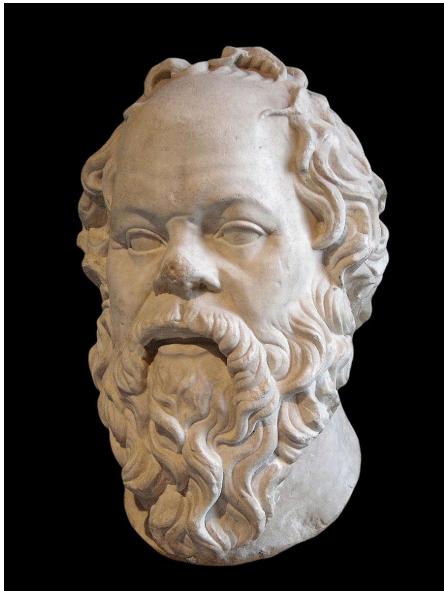
Clear and unambiguous?



A *proof* is a **rigorous mathematical argument** that demonstrates why a given answer is correct

even to the most **serious skeptic**.

Clear and unambiguous?



*54·43. $\vdash \therefore \alpha, \beta \in 1. \text{D} : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54·26. \text{D} \vdash \therefore \alpha = \iota'x . \beta = \iota'y . \text{D} : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

[*51·231] $\equiv . \iota'x \cap \iota'y = \Lambda .$

[*13·12] $\equiv . \alpha \cap \beta = \Lambda \quad (1)$

$\vdash . (1) . *11·11·35 . \text{D}$

$\vdash \therefore (\exists x, y) . \alpha = \iota'x . \beta = \iota'y . \text{D} : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda \quad (2)$

$\vdash . (2) . *11·54 . *52·1 . \text{D} \vdash . \text{Prop}$

From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

(Don't worry, I won't ask you to write this)

Everything is Logic

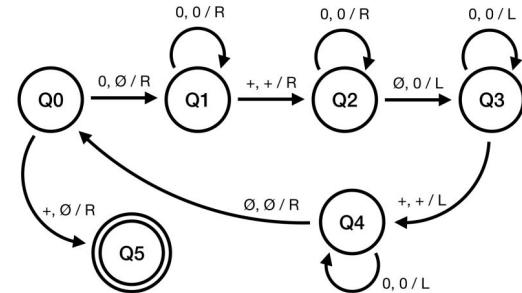


Proofs can be applied to programs!

Programs are mathematical objects

We can test if the program is correct by (again):

1. Writing down what the program **does** (and should do)
2. Coming up with a **rigorous mathematical argument**
3. Using **automatic tools** check (verify) that argument



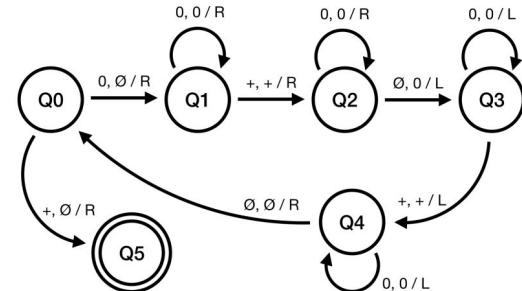
[0, 1, 2, 3, ...]



program verification

Proofs can be applied to programs!

Programs are mathematical objects



[0, 1, 2, 3, ...]

We can test if the program is correct by (again):

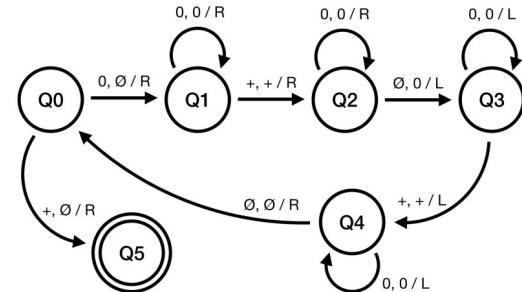
1. Writing down what the program **does (and should do)**
2. ~~Coming up with a rigorous mathematical argument~~
Have the computer come up with a proof!
3. Using **automatic tools** check (verify) that argument



automated verification

Proofs can be applied to programs!

Programs are mathematical objects



[0, 1, 2, 3, ...]

We can test if the program is correct by (again):

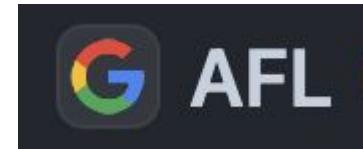
1. Writing down what the program **does (and should do)**
2. ~~Coming up with a rigorous mathematical argument~~
Try running the program on some inputs
3. Using **automatic tools** check (verify) that argument



program testing

Tools used in industry

Testing tools



(Many others)

Tools used in industry

Automated theorem provers



Microsoft
Research

Z3

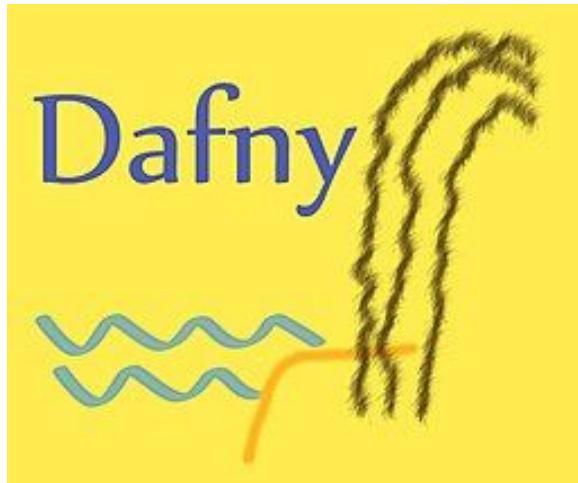
Z3 is a theorem prover from Microsoft Research. It is licensed under the [MIT license](#).



"The total number of invocations of Zelkova ranges from a few million to tens of millions in a single day"

Tools used in industry

Program verifiers



(Originally developed at Microsoft
Research, now hosted by Amazon)



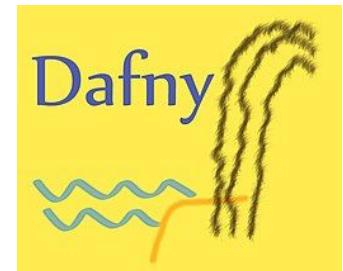
Recap: what this class is about



1. Write down what the program **does** (and should do)
2. Use **automatic tools** check (verify) whether the program meets its requirements



Z3



Plan for today

1. What is this class about?
2. Short discussion/activity
3. FAQ, syllabus, and logistics
4. Demos (time permitting)

FAQ

Q: What is a special topics course?

A: basically an **experimental** course offering



Q: What is a special topics course?

A: basically an **experimental** course offering



Q: What is a special topics course?

A: basically an **experimental** course offering



Listed title: “Programming languages & Compilers” is a default placeholder

Q: Is this class about programming languages and compilers?

Short answer: **Yes** and **No**

Long answer:

Yes: The concepts covered in this course are fundamental to PL and compiler development

No: Not a traditional languages or compilers class (No overlap with 140A or 142)

Q: Are there any prerequisites?

A: Informal prerequisites:

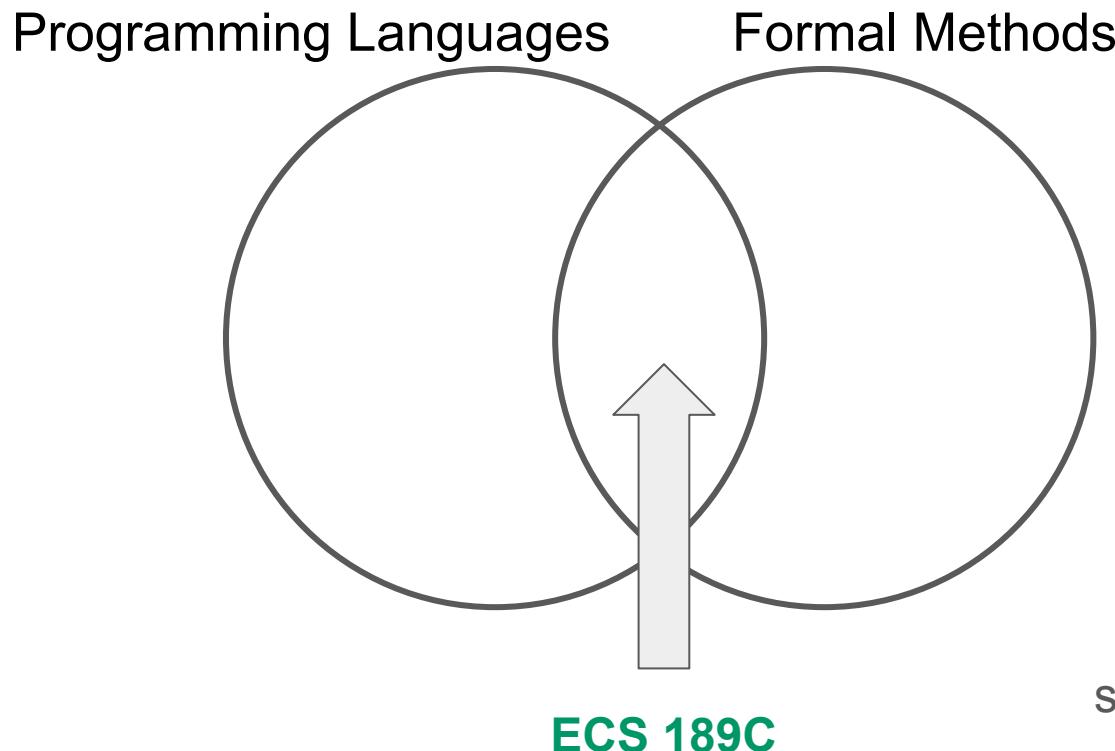
- A prior course in mathematical reasoning (e.g. ECS 20) is also helpful, and generally assumed
- I will assume a basic programming background (e.g., ability to write FizzBuzz, ECS 36A/36B/36C)

Q: Will I be required to write (pen-and-paper) mathematical proofs? (like in ECS 20 or 120)

A: **No**

- We will generally ask the computer to write the proof for us!
- Encourage a **hands-on** approach: the tools we will cover can help you check your work
- If you are interested in advanced development of formally verified software, including delving more into the theory behind it, you can consider my graduate class, ECS 261

Q: What area of computer science is ECS 189C?



Related areas:
theoretical computer
science, SE, security, ...

Q: Is this course right for me?

Short answer: **Probably!**

Long answer: **Especially if:**

- You want to know the fundamental principles of software correctness (specification, testing, and verification) and learn about tools that are used in industry

Syllabus & logistics

Waitlist

During the first/second weeks: Please do attend the class

- Some people will drop
- I'm not allowed to issue any PTAs (department policy)

End of the second week: let me know if you are still on the waitlist

- Deadline to drop: 10th day of the quarter/instruction (drop deadline)
- Depending on interest, I will request to increase the enrollment cap

TL;DR: Please attend the lectures even if you are on the waitlist.

Graded work

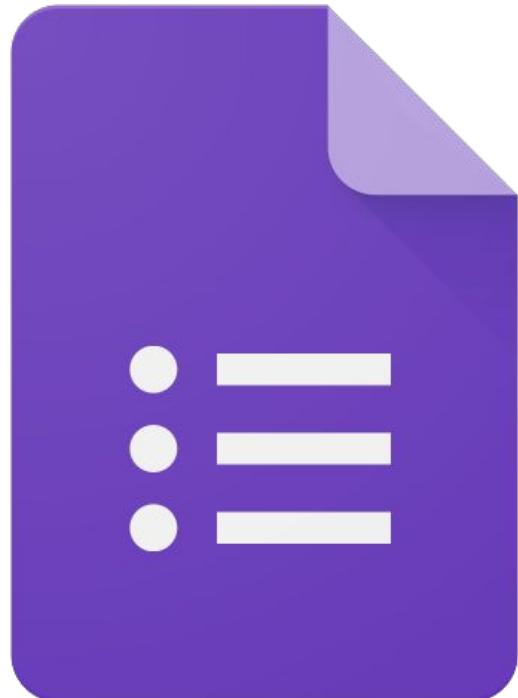
- **Participation (10%)**
- **Homeworks (20%)**
- **Midterm (30%)**
- **Final (40%)**

Attendance and participation (10%)

Fill out the in-class polls (participation points only)

If you are sick: starting Thursday, you may join the class remotely via Zoom (the quality may not be as good)

If you miss class: Lectures are recorded (best-effort). You can make up the in-class polls at any time



Lectures - live coding

Slides and lecture notes posted before/after the lecture! At:

<https://github.com/DavisPL-Teaching/189c>

(course links are pinned in Piazza)

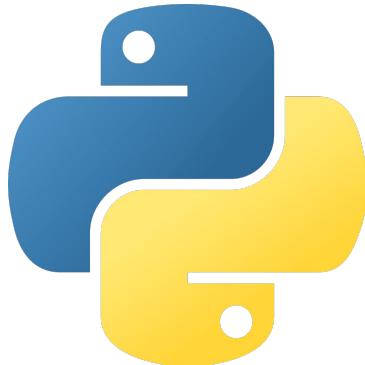
Bring your laptop!

Homeworks (20%)

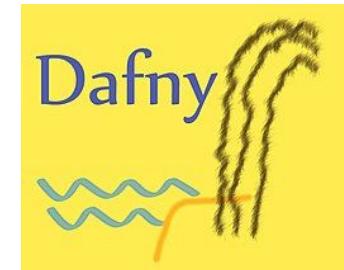
Coding assignments – due roughly bi-weekly

Generally longer form with many small parts

About 5 planned



Z3



Homework 0: installation help



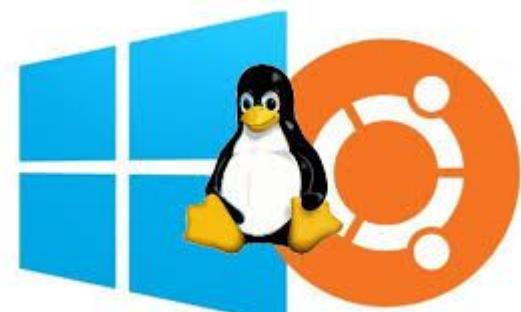
(submission link posted
by next class)



Homework 0: installation help

I recommend using **MacOS or Linux**

If you are on Windows, I recommend



Windows Subsystem for Linux (WSL)

- Well-engineered tool and very useful to know about when doing software development work in the real world!
- <https://learn.microsoft.com/en-us/windows/wsl/install>

Homework Grading

Most important: please run your code

- Software engineering is about running software!
- Running and testing your code frequently is an important part of developing software in the real world – including ensuring your code works on someone else's machine
- **⚠️ Code that doesn't run may receive at most half credit.
Please double check!**
- **⚠️ All HW installation issues should be resolved during HW0**

Exams

- Midterm exam: 30%
- Final exam: 40%
- Mark your calendar: **Friday, March 20, 10:30am-12:30pm**
- Exams are closed book, single-sided cheat sheet

Piazza

Please join and monitor Piazza!

[ECS 189C on Piazza](#)

- Don't email me, post to Piazza!
- Make your post public and (if you prefer) anonymous
- Additional platforms: Canvas and Gradescope

AI Policy

AI use is allowed on homework assignments.

Some advice: AI is a tool - use it wisely!

- Use caution that your use of AI is aiding (and not preventing) your own understanding of the material and critical thinking skills.
- (Current generation) AI has some trouble with Dafny and Z3. Don't fall into the trap of assuming AI is right!
- Exam will be in-class and closed-book
- [Advice from Jason Lowe-Power](#)

Collaboration Policy

- Collaboration is allowed and encouraged!
- You can work on the homeworks in groups of 2-3
- Please list your collaborators at the top of your homework
- Everyone should submit their own solution

A Rough Schedule (subject to change)

We have about 10 weeks, here is a rough plan:

- Weeks 1-2: introduction, writing program specifications, testing with Hypothesis
- Weeks 3-4: automated verification tools, Z3
- Midterm
- Weeks 5-7: interactive verification tools, Dafny, Hoare logic
- Week 8: advanced topics (if time permits)

Communication

TA: Jason Yoo

Office hours: TBD (will be posted on Piazza)

Please use Piazza for questions (not email)

Respect and discrimination

Please be nice!

Include everyone in group discussions

Reach out to me if there are any problems

Late policy

Polls and HW0: can be made up

- You don't need to email me about this, just submit the poll! :)

For HW, I cannot guarantee that late work is graded, but I encourage you to keep working on an assignment if you didn't finish it by the deadline. Gradescope will allow submissions a few days late, and we may choose to grade these late submissions, at our discretion.

- Why no "hard" policy - this is also how it works in the real world :-)
- **Some homeworks are long - please start early!**

Other disclaimers

- [UC Davis Job Scams Prevention](#)
- [Policies against harassment and discrimination](#)
- [List of resources for student health and well-being](#)
- [SISS](#) for international student issues (and reach out to your undergraduate advisors)

Questions for me?

Plan for today

1. What is this class about?
2. Short discussion/activity
3. FAQ, syllabus, and logistics
4. Demos (time permitting)

Demos

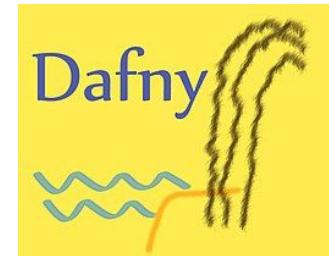
Lecture 0 Demos

Learning objectives

- Understand the concept of software correctness and its importance.
- Use random testing tools like Hypothesis for software testing.
- Use model-based verifiers like Z3 for software analysis.
- Apply program verification tools such as Dafny to ensure the correctness of software.
- Explore advanced topics in software correctness, such as advanced type systems, safety and liveness properties, and program logics.



Z3



✨ Puzzle ✨

I'm thinking of 2 numbers.

The +, *, -, and / of the numbers are (not necessarily in this order):

20, 95, 105, 500

What are the numbers?

✨ Puzzle ✨

(Another one)

The +, *, -, and / of the numbers are (not necessarily in this order):

2, 6, 18, 72

What are the numbers?

✨ Puzzle ✨

(Another one)

The +, *, -, and / of the numbers are (not necessarily in this order):

2, 6, 18, 72

Is this always possible?