ECS 189C: Software Correctness

(Special Topics in Programming Languages & Compilers)

Lecture 0 – SQ2024

Welcome!

- This is a **special topics course**
- Flyer
- Cap: 60 BUT some will drop – more expected off waitlist
- 4 units

⚠️ **Please remember to change the number of units to 4**

About the Instructor

What I do: Programming Languages

(Started at Davis: July 2023)

Website

DavisPL Research Group

Plan for today

1. What is this class about?
2. Syllabus and logistics
3. Demo

What is this class about?

Example

Please build me a car

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

Example

Please build me a car

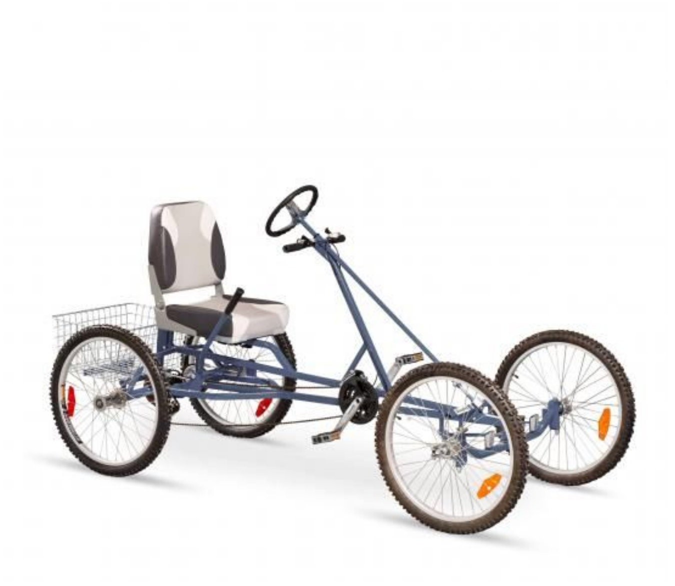
It should have four wheels and you can drive it places

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

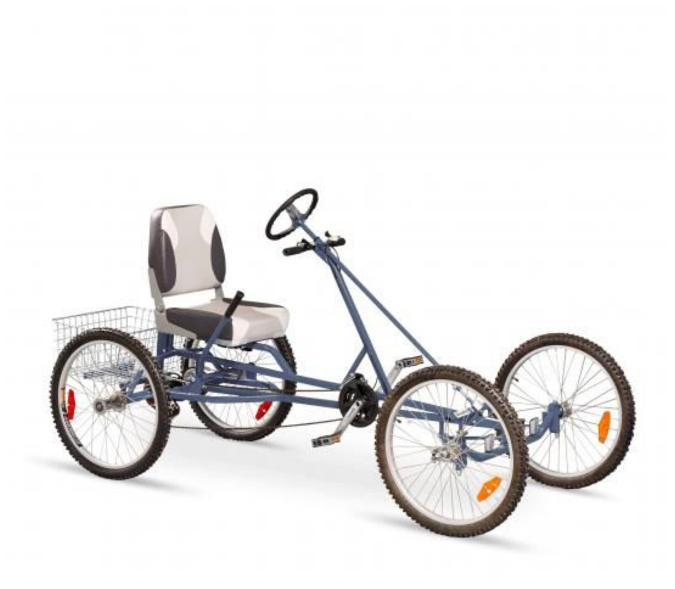


Ok, here you go

Example

Please build me a car

… it should also have a roof and seats inside

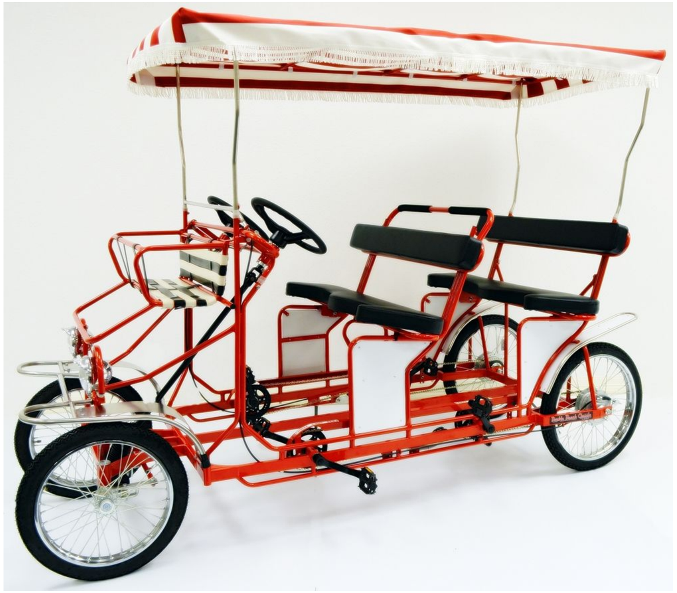


Ok, here you go

Example

Please build me a car

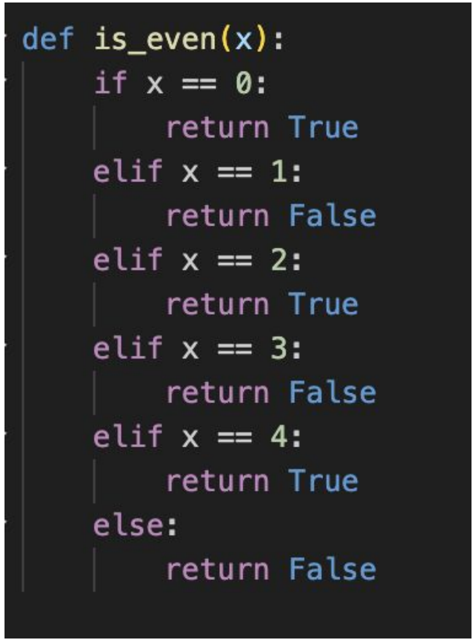
… it should also have a roof and seats inside



Ok, here you go

What does this have to do with programming?

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

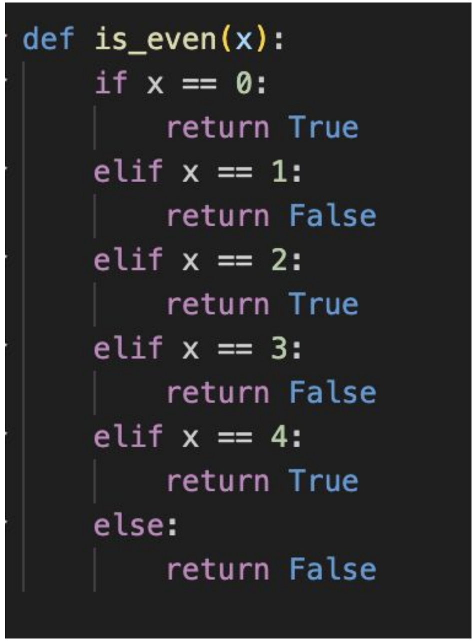


```
def is_even(x):
    if x == 0:
        return True
    elif x == 1:
        return False
    elif x == 2:
        return True
    elif x == 3:
        return False
    elif x == 4:
        return True
    else:
        return False
```

Ok, here you go

What does this have to do with programming?

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



```python
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

Is this correct?

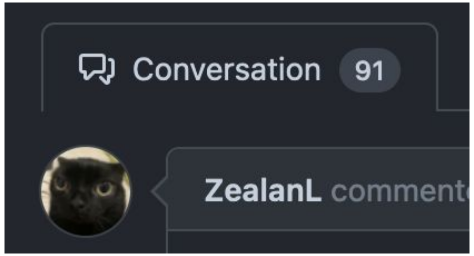
(Why not?)

What does this have to do with programming?

Please write a program to play chess

Ok, here you go

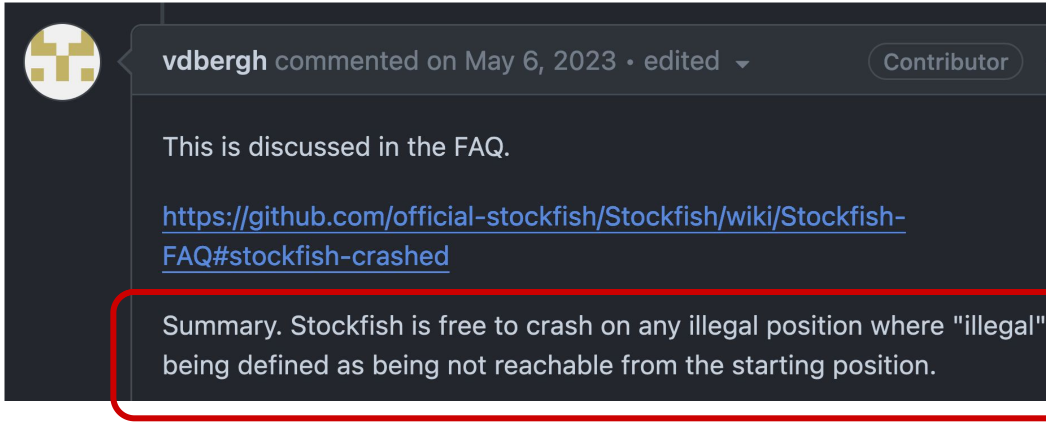
What does this have to do with programming?



Conversation 91

ZealanL comment

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



vdbergh commented on May 6, 2023 • edited

Contributor

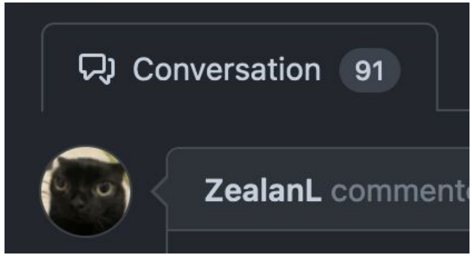
This is discussed in the FAQ.

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

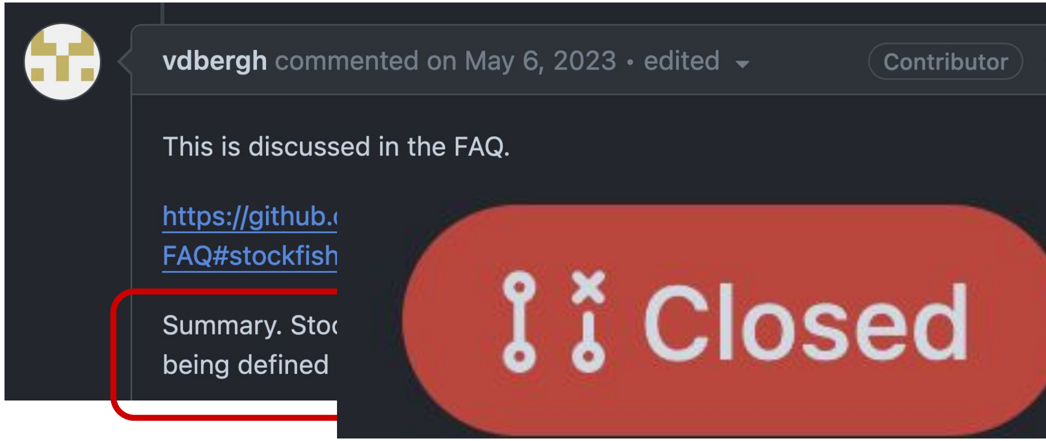
What does this have to do with programming?



Conversation 91

ZealanL comment

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



vdbergh commented on May 6, 2023 • edited

Contributor

This is discussed in the FAQ.

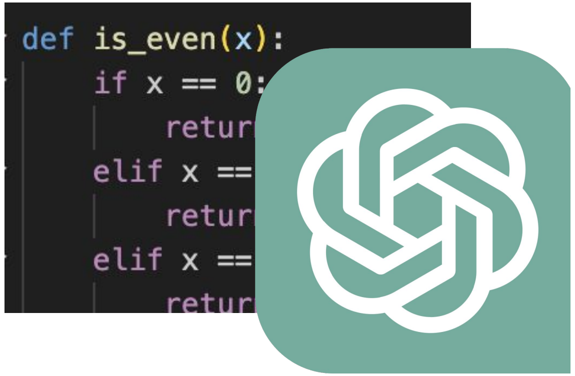
https://github.
FAQ#stockfish

Summary. Stoc
being defined

It’s a feature, not a bug

What does this have to do with programming?

Please write a program to <do some task>



```
def is_even(x):
    if x == 0:
        retur
    elif x ==
        retur
    elif x ==
        retur
```

Ok, here you go

Is this correct?

(Why not?)

What does this have to do with programming?

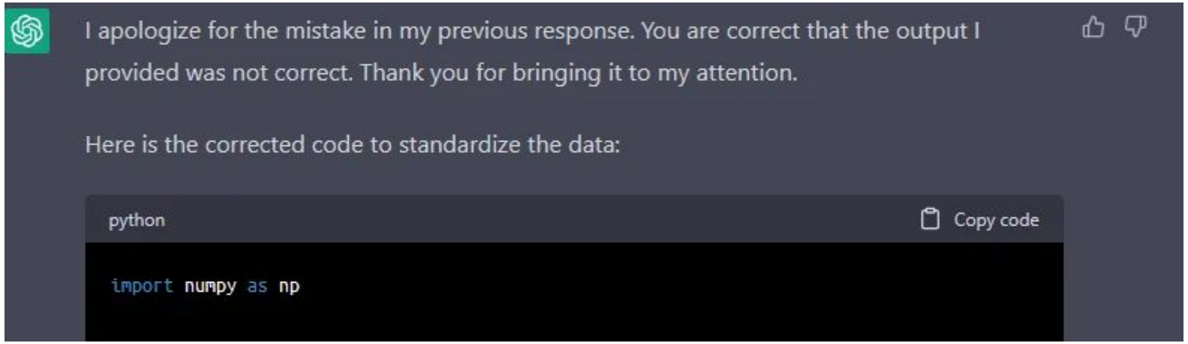
Please write a program to <do some task>

```
def is_even(x):
    if x == 0:
        retur
    elif x ==
        retur
    elif x ==
        retur
```

Ok, here you go

Is this correct?

(Why not?)



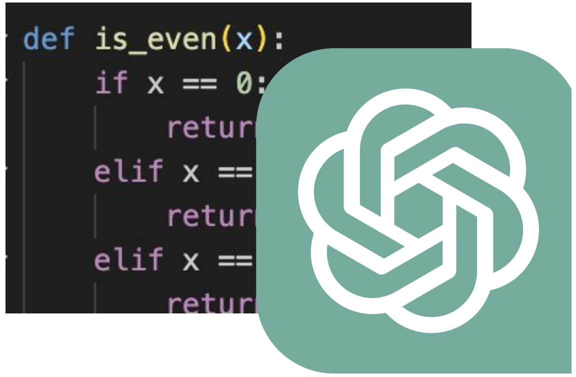
I apologize for the mistake in my previous response. You are correct that the output I provided was not correct. Thank you for bringing it to my attention.

Here is the corrected code to standardize the data:

```python
import numpy as np
```

What does this have to do with programming?

Please write a program to <do some task>

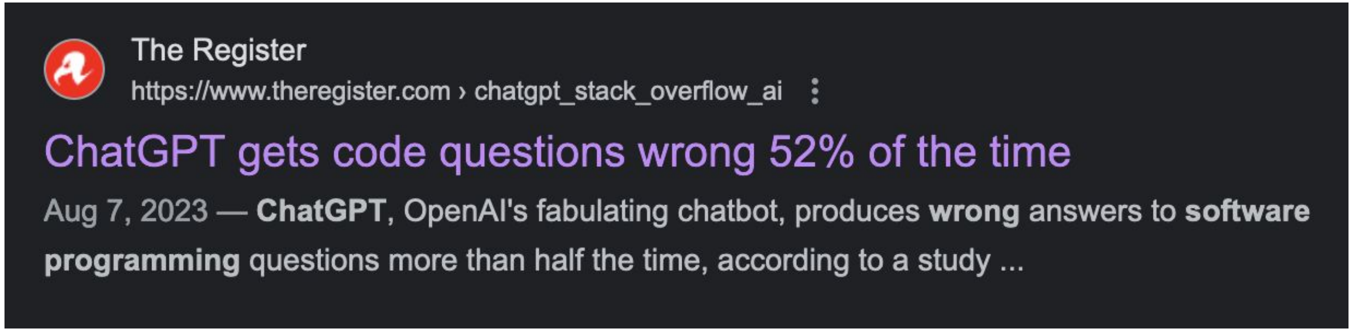


```
def is_even(x):
    if x == 0:
        retur
    elif x ==
        retur
    elif x ==
        retur
```

Ok, here you go

Is this correct?

(Why not?)



The Register
https://www.theregister.com › chatgpt_stack_overflow_ai

ChatGPT gets code questions wrong 52% of the time

Aug 7, 2023 — **ChatGPT**, OpenAI's fabulating chatbot, produces **wrong** answers to **software programming** questions more than half the time, according to a study ...

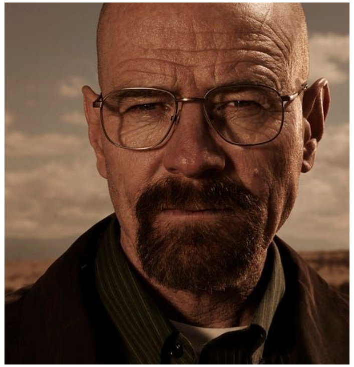
Making the situation worse…

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

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

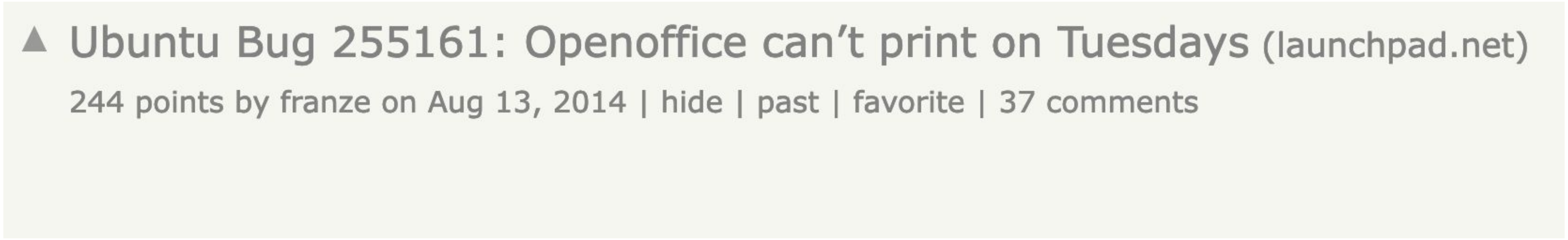
Or fail to show up at all.

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



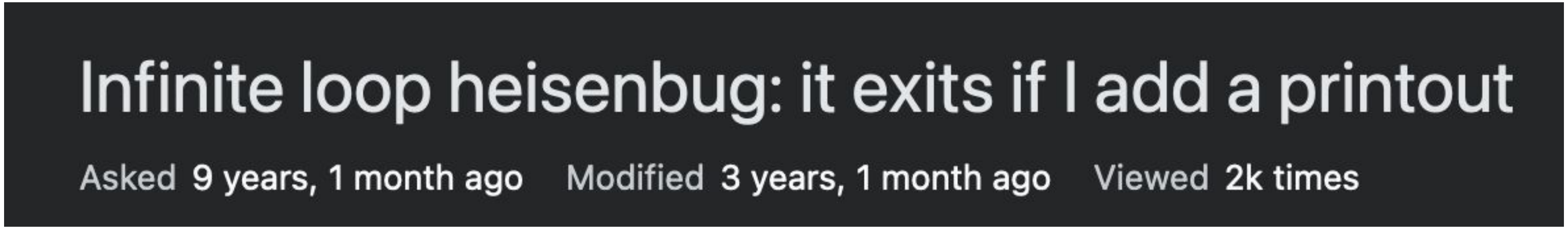
Heisenbugs

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 franze 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

The problem

We need a

clear and unambiguous

way to determine if programs are correct.

The problem

We need a

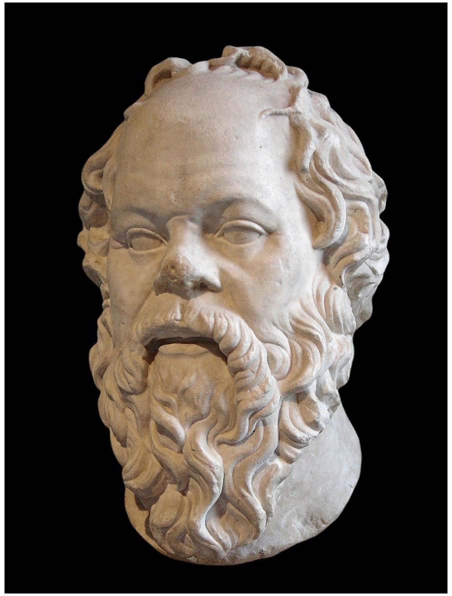
clear and unambiguous

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?



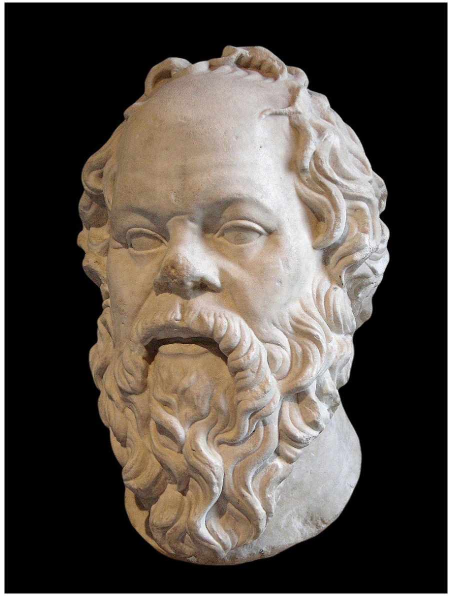
All men are mortal.

Socrates is a man.

Therefore, Socrates is mortal.[2]

Logical Syllogism

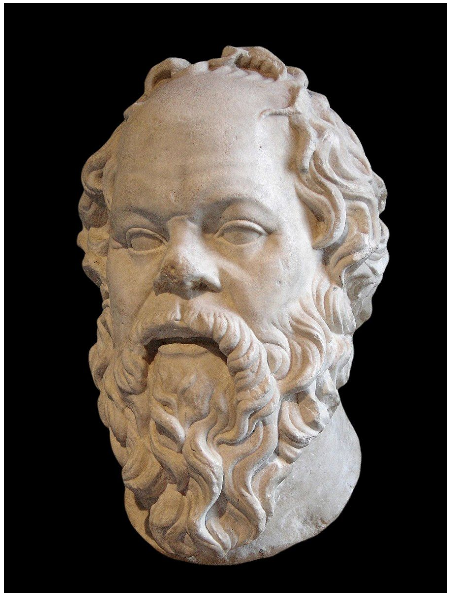
Clear and unambiguous?

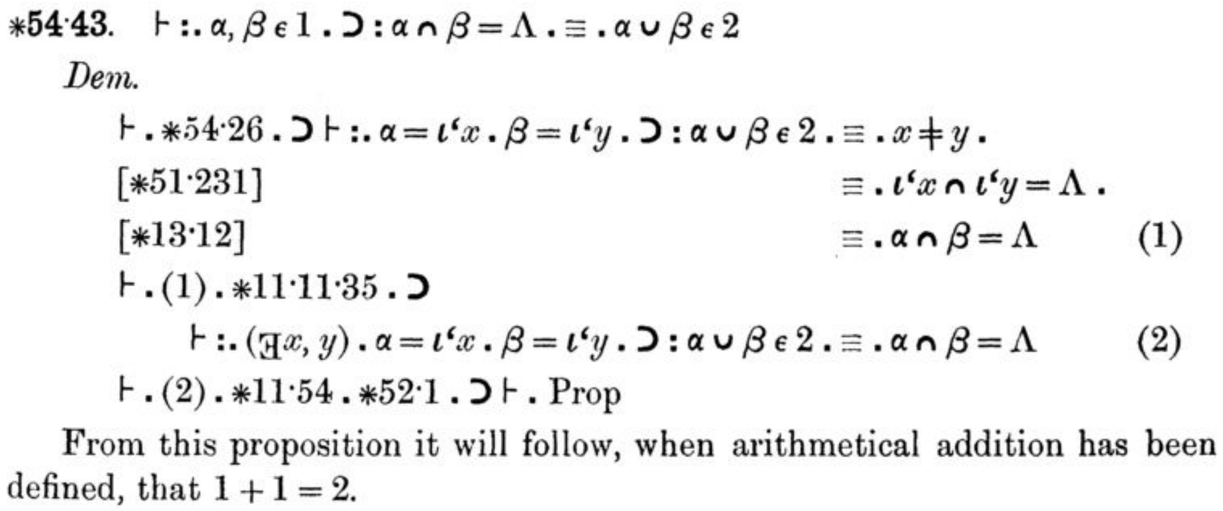


A *proof* is a rigorous mathematical argument that demonstrates why a given answer is correct

even to the most serious skeptic.

Clear and unambiguous?





$*54\cdot43.\ \vdash :.\ \alpha, \beta \in 1 \,.\supset :\ \alpha \cap \beta = \Lambda \,.\equiv.\ \alpha \cup \beta \in 2$

Dem.

$\vdash .\ *54\cdot26 \,.\supset \vdash :.\ \alpha = \iota\text{‘}x \,.\ \beta = \iota\text{‘}y \,.\supset :\ \alpha \cup \beta \in 2 \,.\equiv.\ x \neq y \,.$

$[*51\cdot231] \qquad \equiv.\ \iota\text{‘}x \cap \iota\text{‘}y = \Lambda \,.$

$[*13\cdot12] \qquad \equiv.\ \alpha \cap \beta = \Lambda \qquad (1)$

$\vdash .\ (1) \,.\ *11\cdot11\cdot35 \,.\supset$

$\vdash :.\ (\exists x, y) \,.\ \alpha = \iota\text{‘}x \,.\ \beta = \iota\text{‘}y \,.\supset :\ \alpha \cup \beta \in 2 \,.\equiv.\ \alpha \cap \beta = \Lambda \qquad (2)$

$\vdash .\ (2) \,.\ *11\cdot54 \,.\ *52\cdot1 \,.\supset \vdash .$ Prop

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

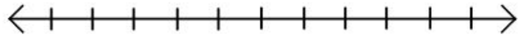
(Don’t worry, I won’t ask you to write this)

Everything is Logic

Proofs can be applied to programs!

Programs can be modeled as mathematical objects

[0, 1, 2, 3, …]



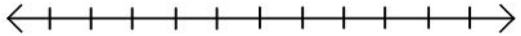
We can test if the program is correct by:

1. Writing down what the program does (and should do)
2. Coming up with a rigorous mathematical argument

Proofs can be applied to programs!

Programs can be modeled as mathematical objects

[0, 1, 2, 3, …]



We can test if the program is correct by:

1. Writing down what the program does (and should do)
2. Coming up with a rigorous mathematical argument

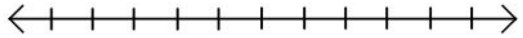
Aside: proofs ARE programs (Curry-Howard 1934, 1969)

Curry-Howard Correspondence

Proofs can be applied to programs!

Programs can be modeled as mathematical objects

[0, 1, 2, 3, …]



We can test if the program is correct by:

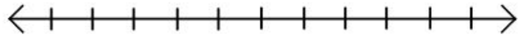
1. Writing down what the program does (and should do)
2. ~~Coming up with a rigorous mathematical argument~~

~~Aside: proofs ARE programs (Curry Howard 1934, 1969)~~

~~Curry-Howard Correspondence~~

Proofs can be applied to programs!

Programs can be modeled as mathematical objects [0, 1, 2, 3, …]



We can test if the program is correct by:

1. Writing down what the program does (and should do)
2. ~~Coming up with a rigorous mathematical argument~~

This class is not about writing proofs
Better: we will make the computer do the math for us!

✨ Puzzle ✨

I’m thinking of 4 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):

3, 10, 20, 75

What are the numbers?

✨ Puzzle ✨

(Another one)

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

3, 10, 20, 75

Is this always possible?

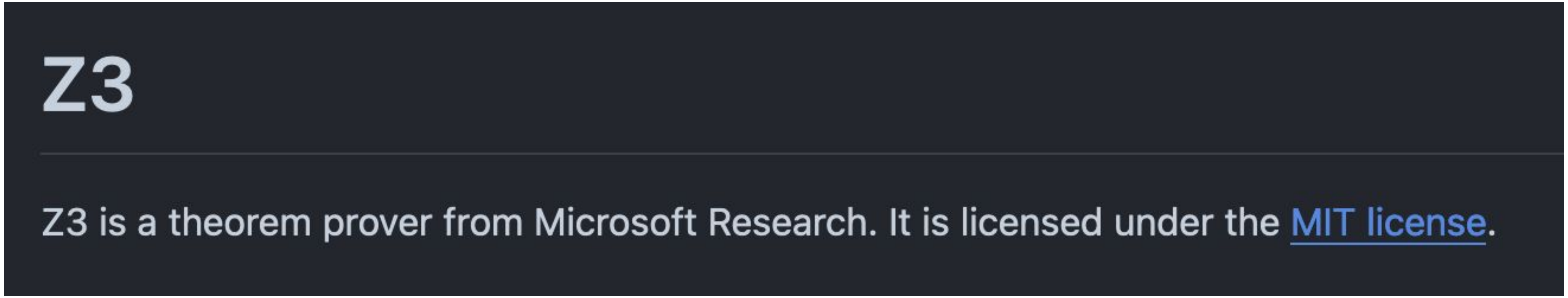
Tools used in industry

Testing tools

(Many others)

Tools used in industry

Automated theorem provers



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

Idris

Agda

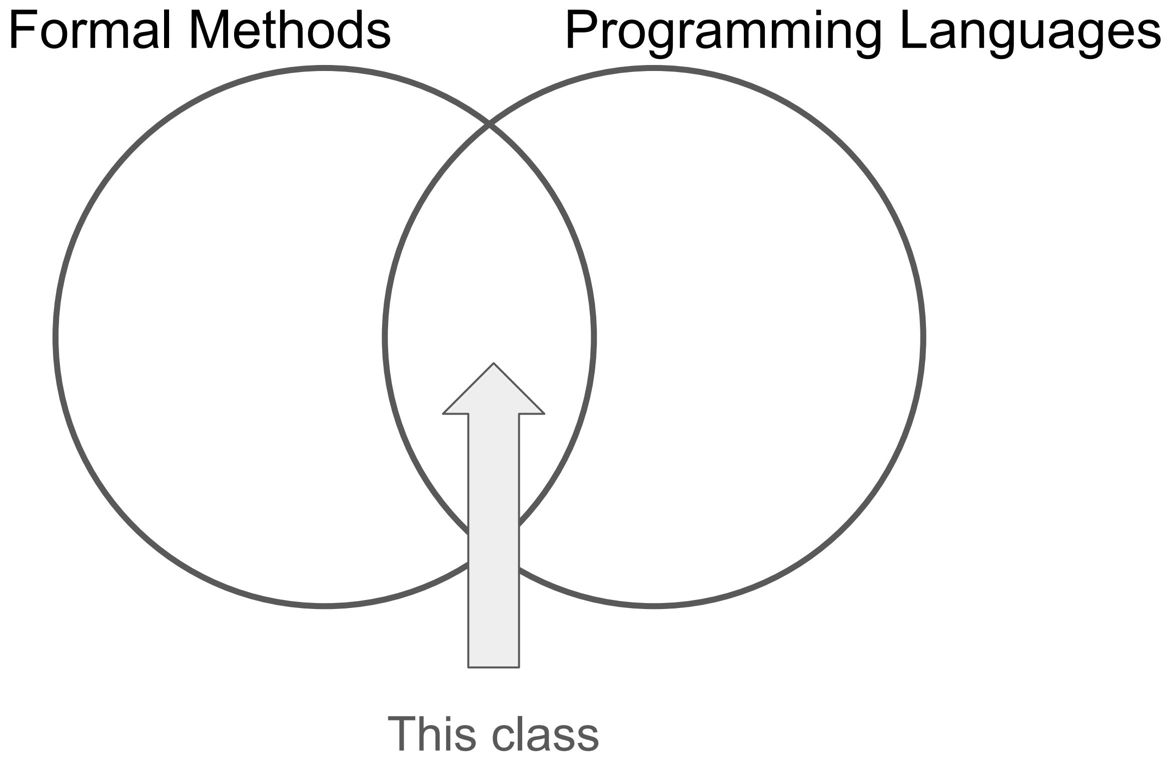
Dafny

Tools used in industry

This class is not about writing proofs

- I am more interested in:
    - showing you the **industrial tools available** that can be used to practically **test and/or demonstrate** software correctness.
    - helping you **think clearly about** what programs you write, why they work, and why they don't

Areas of computer science



Summary: what this class is about

Concepts and **tools** that help you understand:

1. What the software is;
2. What it is supposed to do; and
3. Why it works (or why it doesn’t)

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?

Short answer: No

Long answer:

- I will assume a basic programming background
  - (e.g., ability to write FizzBuzz, ECS 36A/36B/36C)
- Some familiarity with mathematical reasoning (e.g. ECS 20) helpful, but not required

Q: Is this course right for me?

Short answer: Probably!

Long answer: Especially if:

- You want to know the fundamental principles of software correctness and learn about tools that are used in industry
- You are interested in thinking mathematical or logically about software and what it does

Plan for today

1. What is this class about?
2. Syllabus and logistics
3. Demo

Learning objectives

1. Understand the concept of software correctness and its importance.
2. Use random testing tools like Hypothesis for software testing.
3. Use model-based verifiers like Z3 for software analysis.
4. Apply program verification tools such as Dafny to ensure the correctness of software.
5. Explore advanced topics in software correctness, including advanced type systems and concurrency.

Waitlist

During the first week: Please do attend the class

- Some people will drop
- Unfortunately, I can’t issue any PTAs

By the end of the second week: You will be admitted off the waitlist if you have been attending class

- Deadline to drop: 10th day of the quarter/instruction (drop deadline)

Waitlist

During the third week:

- Enrollment should already be determined by this time
- Official add deadline is ~13th day
- You must be enrolled to continue participating in the course

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

Grading

- **Homeworks (50%):** 5 assignments (due bi-weekly).
- **Participation (10%):** via in-class polls and Piazza.
- **Final Exam (40%):** covering all topics covered in the course.

Grading

- **Homeworks (50%):** 5 assignments (due bi-weekly).
- **Participation (10%):** via in-class polls and Piazza.
- **Final Exam (40%):** covering all topics covered in the course.

Homework 0 (setup + installation) will be due Monday 1 week from today

Homework 1-5 every ~2 weeks

Attendance

There are in-class polls (participation points only)

If you are sick: Starting from lecture 2, you may join the class remotely via Zoom (the quality may not be as good)

If you miss class: Lectures will be recorded. There will be a way to make up the in-class polls

Polls – Poll Everywhere

Piazza

189C

AI Policy

- Allowed and encouraged! (But not required)
- Advice from Jason Lowe-Power
- Final exam will be in-class and closed-book

Collaboration Policy

- Allowed and encouraged!
- Please list your collaborators at the top of your homework

Schedule

Schedule

Communication

TA: **Parnian Kamran**

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

Questions for me

Demos

Lecture 0 Demos