

Week 02 Studio Intro / Elements of Programming

**CS1101S AY21/22 Semester 1
Studio 05E**

16 Aug 2021

Yan Xiaozhi (David)
@david_eom
yan_xiaozhi@u.nus.edu





NUS School of Computing

13
COMPUTING
DRIVE

Who is this dude?

- Yan Xiaozhi (David)
- Year 2 Computer Science
- ~0 programming experience before entering university
- Tembusu College, tShuttlers
- Guitar, piano, badminton
- Fun fact: I had 3 nationalities at one point LOL
- Summer internship at IMDA, working on TTS with Singaporean accent
- A huge troll



Avenger?

- I am your friend / senior in SoC!
 - Hey, I just took this mod 1 year ago only
 - Time flies...
- I will try my best to guide you through!
 - “I am here for you” - Prof Martin’s education paradigm
 - Not a “teacher”, this is my first time teaching CS as well!
 - Learning process for all of us
 - I might be wrong, but we get to discuss during studio!



What about you guys!

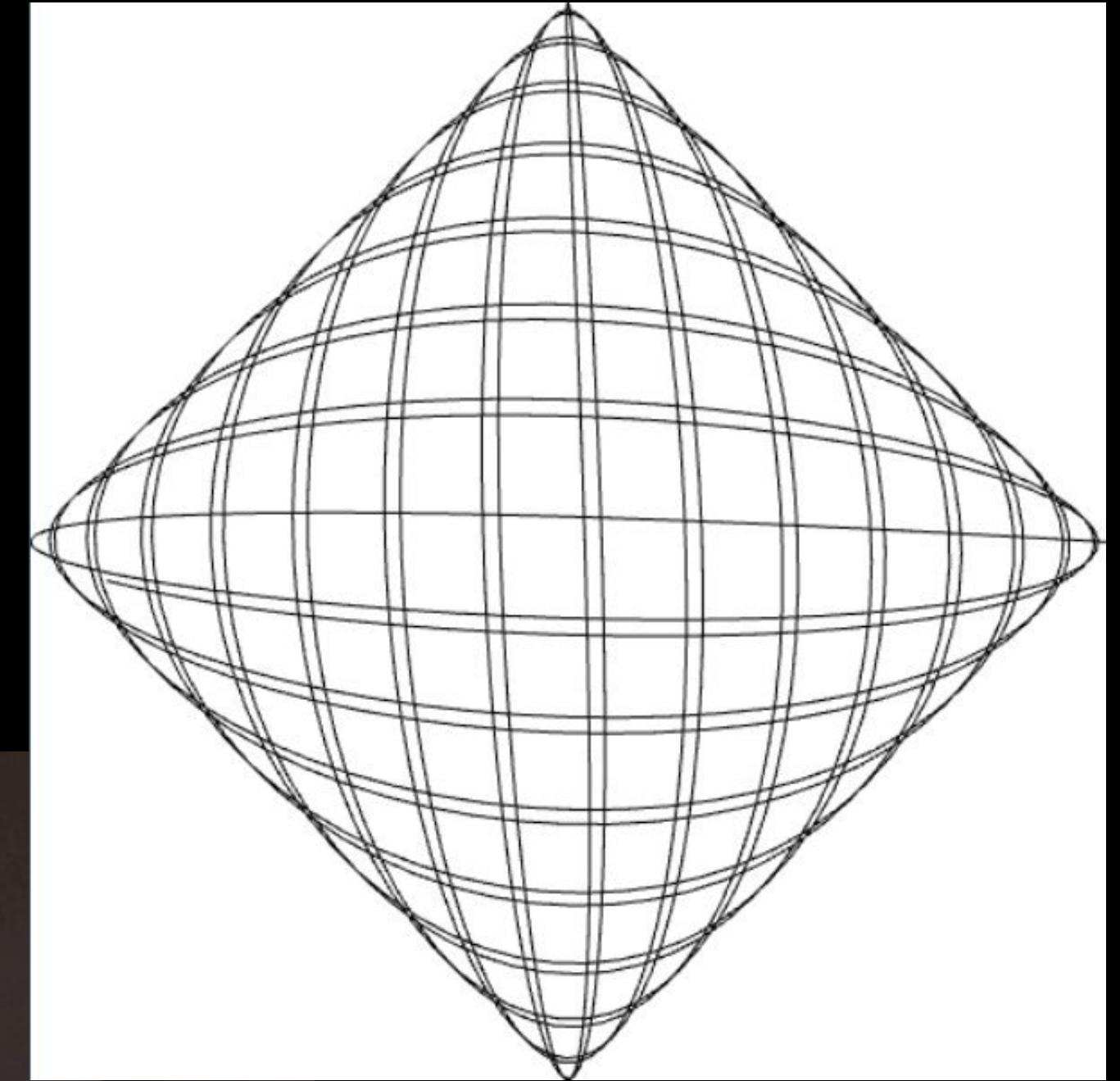
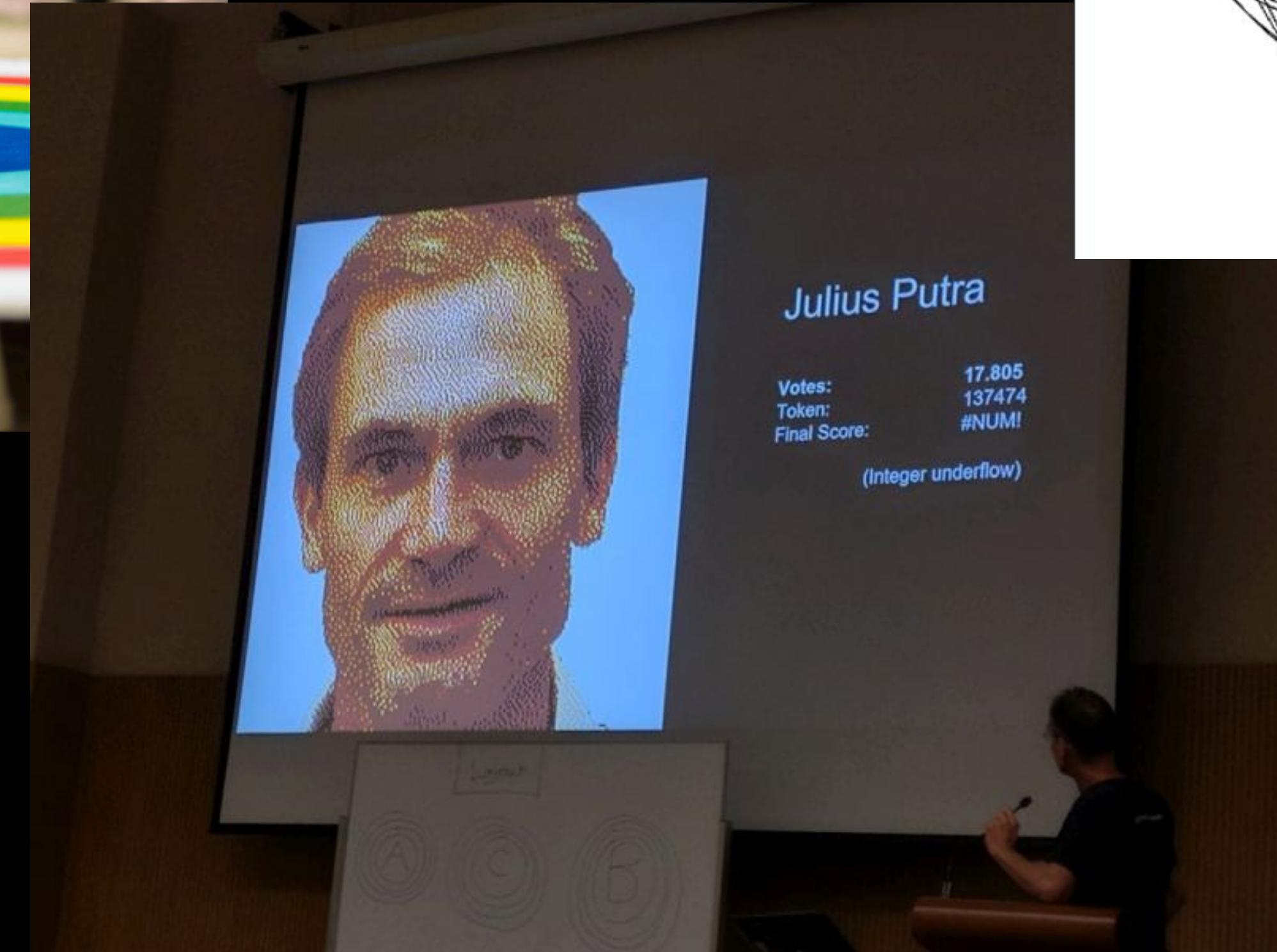
- Name (any preferred name?)
- Any prior programming experience?
- Hobby / fun fact?
- Why CS?
 - ~~(more like why did you choose to torture yourself)~~



What is CS1101S?

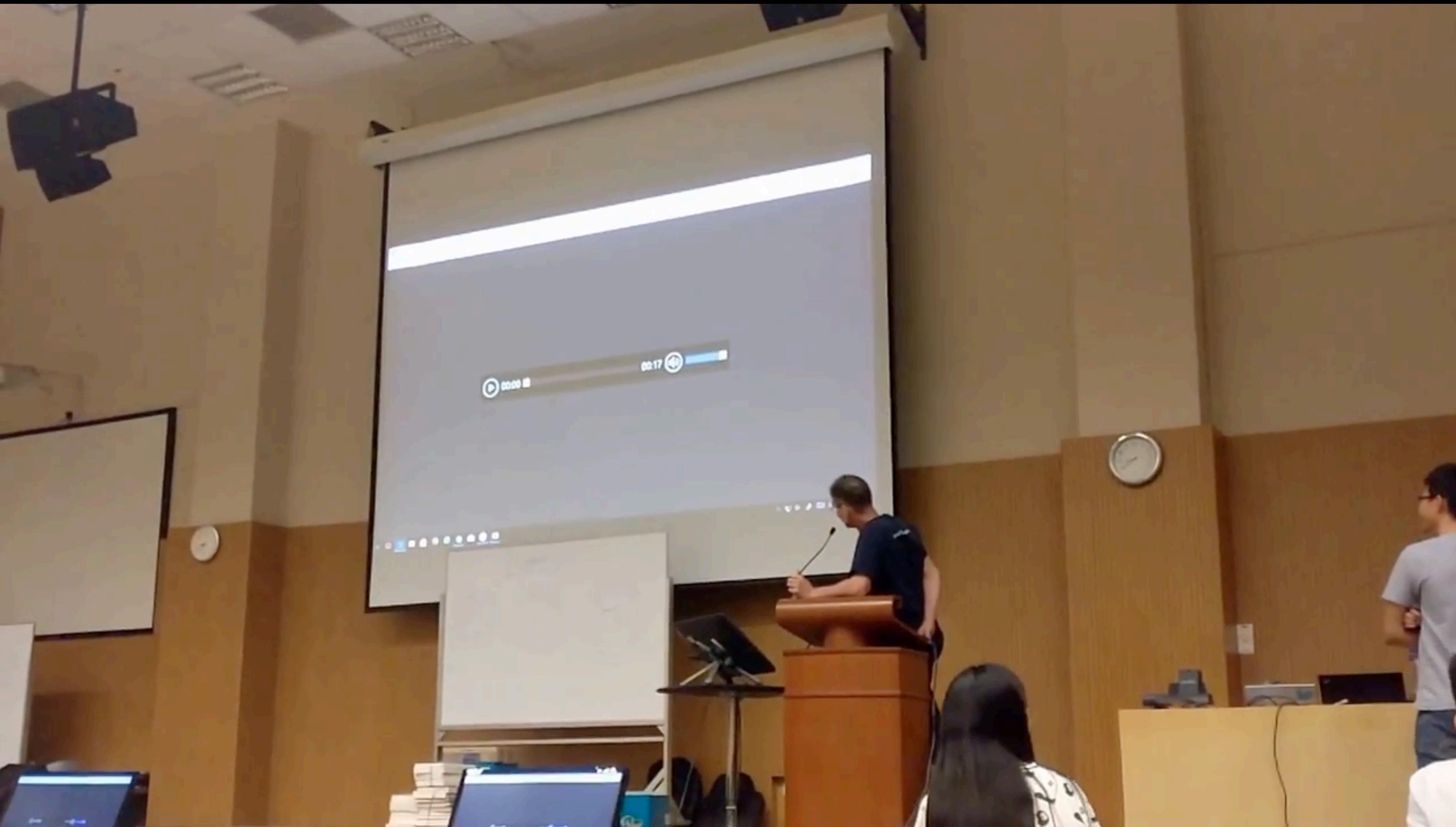
- Introductory module (CS2030S, CS2040S and stuff)
- Workload can be quiiiiite heavy, used to be a 5MC module
- NOT a module to learn programming language or cXding
- *Computational thinking*, how to solve problems
- A VERY fun mod!
 - (Trust me it might be a downhill after 1101S...)

What is CS1101S?



Credit: Julius Putra, CS1101S Discussion Group AY20/21

What is CS1101S?



Got questions?

- Our studio Telegram chat
- Piazza forum
- Other CS friends
- PM me (only as a last resort (:)!
 - Please understand 🙏 Y2 things...
 - I'll reply in the Telegram chat as well!
 - Might be thought-provoking and important for all!
 - No such thing as a stupid question



Expectations

- Attempt the worksheets before coming to studio!
- Discussion-based, pair works
- Speak up in class! Communication is important
- Be punctual! Kindly drop me a message if you are coming late
- Any lesson before / after ?
- Plagiarism

Admin

- Mon, 4pm-6pm, release ~1745 (remind me if i run overtime)
- Break in between
- Recap of content, go through questions, studio worksheet, in-class worksheet
- Before midterm / final: go through past year questions
- Mission / quest grading
 - Will try to finish within 24 / 48 hrs
 - Resubmission: can is can, but don't misuse!

Admin

- EXP: 18% of your grade, most people will get full 18%, CS1010R
- Mission, path, contests ...
- Studio EXP
 - XP is from 0-500
 - Existence ("warm body") is 200 XP baseline
 - 300-350 XP: talk, ask question, contribution
 - 400 XP is exceptional contribution
 - 500 XP: you are doing my job!!
- Studio effort (2%)

5 cents from me

- Find friends to discuss and learn with
- Ask questions frequently, Piazza can be quite useful!
- Learn to use the source acad well! e.g. shortcuts, stepper etc.
- Prioritise your time
- Reading Assessments - DON'T OVERLOOK THEM!
- Plan, then programme! Try not to just run it over and over again...
- Give the quests a try! (sometimes can be easier than mission)

**Computer science is no
more about computers
than astronomy is about
telescopes.**

- Edsger W. Dijkstra



Recap

Elements of Programming

Primitive Expression

- Numerals: 1, 2, 3
- Primitive expressions are not programmes
- Addition of semicolon makes it into a programme
 - e.g. 123;

Elements of Programming

Operator Combinations

- Infix notation with precedences
 - Infix: operator between operands (vs prefix, postfix)
 - Precedences: () > * / > + -
 - $(2 + 4 * 6) * (3 + 12)$
- Unary, binary, ternary
 - ! , -
 - + , - , * , / , %
 - ? :

Elements of Programming

Naming Abstraction

- Name declaration
 - Pre-declared names
 - Constants, e.g. `math_PI` / Functions, e.g. `math_sqrt`
 - Constant declaration
 - `const size = 2; 5 * size;`
 - `const`: adds a name-value entry to the table
 - Parameters
 - Function name

Elements of Programming

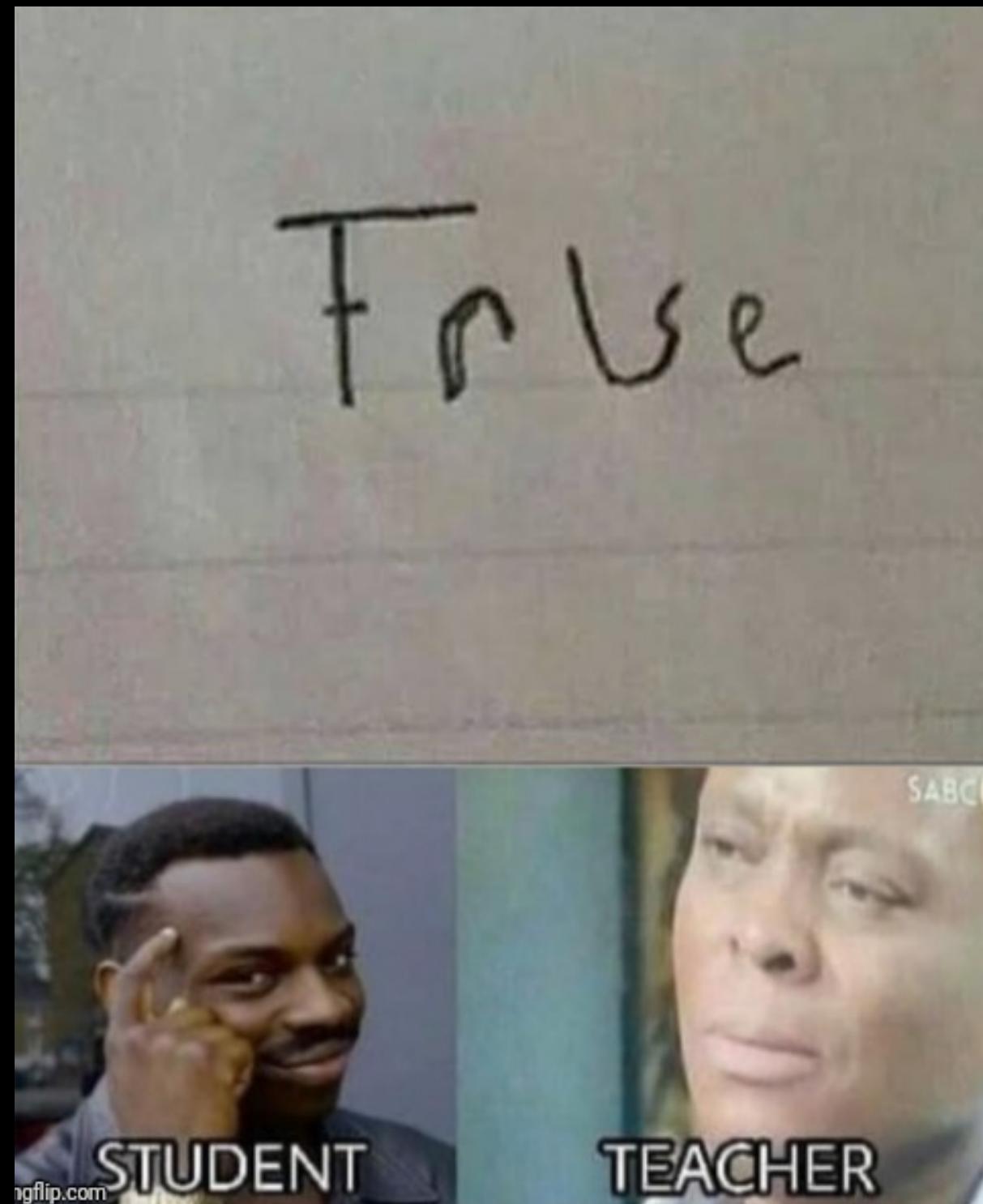
Functional Abstraction

- Parameter vs argument
- `function name(parameters) { return expression; }`
- `function add_one(x) {
 return x + 1;
}`
- `add_one(3);`
- `add_one(add_one(10));`

Elements of Programming

Predicate and Conditional Expressions

- Boolean values: true / false
- Predicate: function or an expression that returns or evaluates to a boolean value
 - `true;`
 - `false;`
 - `x >= 1;`
 - `y === 10;`



Elements of Programming

Predicate and Conditional Expressions

- `predicate ? consequent : alternative;`
- `function absolute(x) {
 return x > 0
 ? x
 : x * -1;
}`

Any Questions?