17th Aug 2020

Studio 2
Intro & Elements of Programming

CS1101S AY20/21 SEM 1
Studio 03A

Chen Xihao
Year 2 Computer Science

chenxihao@u.nus.edu
@BooleanValue

This is me

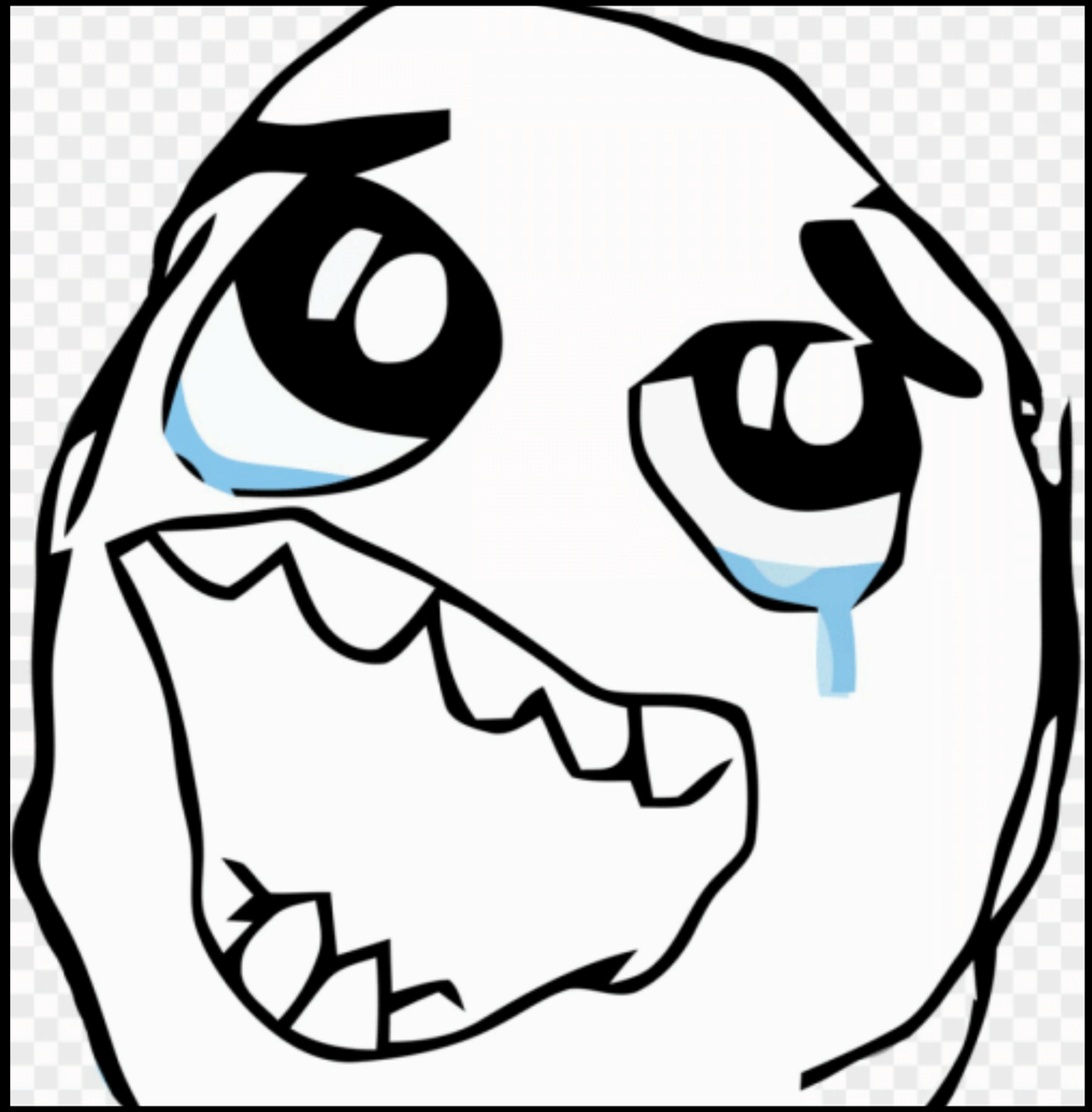
- Chen Xihao
- Year 2 Computer Science
- Some programming experience before university
- Coffee addict
- Varsity Fencer
- ~~Definitely a joker~~

But who am I to you?
My role as an Avenger

- Firstly, your friend!
  - I only took this mod one year ago
  - I still remember the ~~pain and suffering~~ fun I’ve had
- Secondly, I’m here to guide you
  - I’m not a teacher
  - Just here to guide you and give you tips!

How ‘bout u?

- Name
- Programming experience?
- Hobby
- Why ~~torture yourself in~~ CS



Some Facts on this Module

- Rather heavy workload (the 10 hours per week is simply BS)
  - Wasn’t expecting to put in so much effort for an entry level mod -.-||
- Fun fact: CS1101S was a 5MC mod in AY17/18 and before
- We will learn computational thinking and processes, not programming
  - Programming is simply a tool of expression
- Can be tough even for the experienced programmer
- Fun mod! (hopefully)

credits: CS1101S website, Christian Welly's slides

BELL CURVED?

I don’t think so!

Seniors’ Words

“1101s can be *quite challenging* if you have no coding background. Just try your best to complete all the missions on your own. You can ask your friend about the logic but it is better to write the code by yourself.

This mod can be quite time consuming but the time spent will be worth. Don’t memorise the lecture notes, instead, ***understand the logic*** **and the code.”**

- Ding Youjia

“I think I wasn’t that motivated in year 1 sem 1, *everything was a challenge*, but I had fun doing the projects and it really taught me a lot!

Never give up, *work and discuss with your friends about possible solutions*! Together you can get through it!”

- Ee Liang

“Seriously speaking would be to *ignore competition* HAHAHAHA cuz there’s always gonna be someone better than you, what you must do is to stand out and not to be too worried about that in the first sem.”

- Nikhila

In summary: “it’s hard”

But it’s okay, I’m here to see you through to the end!

Expectations in Studio

- Discussion oriented, every one of you will speak
- Attempt the studio questions before coming
- You will present the solutions and I'll guide you on the right path
- Questions? Ask!
- Admin:
  - Studios should end 20 min before official timing (1340 hrs). Remind me if I forget!
  - If you are going to be late, pls inform me so I know when I should start

Expectations in Studio

- Questions outside of class
  - Try asking in the group chat!
- Questions regarding source academy (missions / quests)
  - Discuss with your friends first!

- Pleas bear with me if I can’t answer your questions in class sometimes .-. I need some recapping of the content too, but I’ll definitely get back to you (remind me if I forget)

Other Admin Stuff

- Attendance: Telegram bot attendance (?)
- Missions / quests:
  - Attempt every mission! Do quests only if you have time!
  - Un-submission: please message me
    - don’t abuse this ‘feature’, I need to study too .-.

Word of advice from me

- Time management is crucial!
- Think first, then programme!!!
  - Draw diagrams, get messy! Don’t be afraid to make mistakes!
  - Discuss with your peers
  - Trial and error is the way to go!
- Give the quests a try, but only if you have time.
  - They are very time consuming! (I think some are on the edge of the syllabus)
  - But definitely good practice for problem solving
- ‘Coding’ isn’t the aim of CS, and definitely not this module!

“Computer science is no more about computers than astronomy is about telescopes.”

- Edsger W. Dijkstra

Any questions?

To-do list
(reminder for myself)

- Distribute 3D glasses
- Attendance
- Telegram group https://tinyurl.com/01s03a
- GitHub page https://tinyurl.com/01s03aMaterials

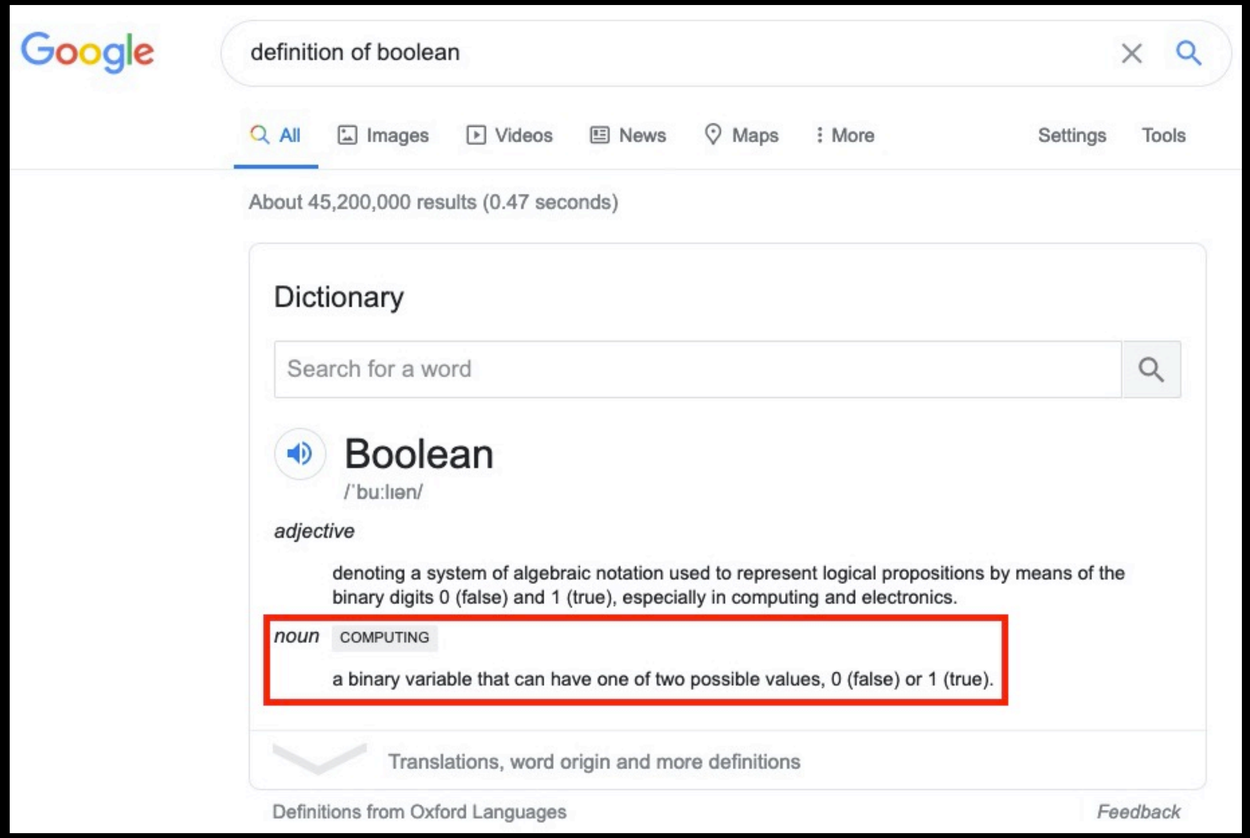
- Time Check!

End of Introduction

Recap

Elements of Programming
Terminology

- Primitives:
  - Numbers: 0, 42, 3.1415926…
  - Booleans: `true` and `false`
    - Only two possible states



Elements of Programming

Terminology

- Statements:
  - constant declaration: `` `const zero = 0;` ``
    - `const` is a ‘keyword’, tells interpreter to add key-value pair into the env
  - function declaration: `` `function foo() { return 0; }` ``
    - `function` is also a ‘keyword’
  - expression statement: `` `2 * zero;` ``

Elements of Programming

Terminology

- Programme: a series / sequence of statements and expressions
- Expression: turned into programmes with `;`
  - primitive number expression:
  - primitive boolean expression:
  - conditional expression:
  - <many more>

Elements of Programming

Terminology

- Operators
  - Unary operator
    - Performed on one operand
    - e.g. `` `!true` ``, `` `-10` ``
  - Binary operator (Source and JS use infix notations)
    - Requires one operand on each side
    - e.g. `` `x + 1` ``, `` `bool1 && bool2` ``

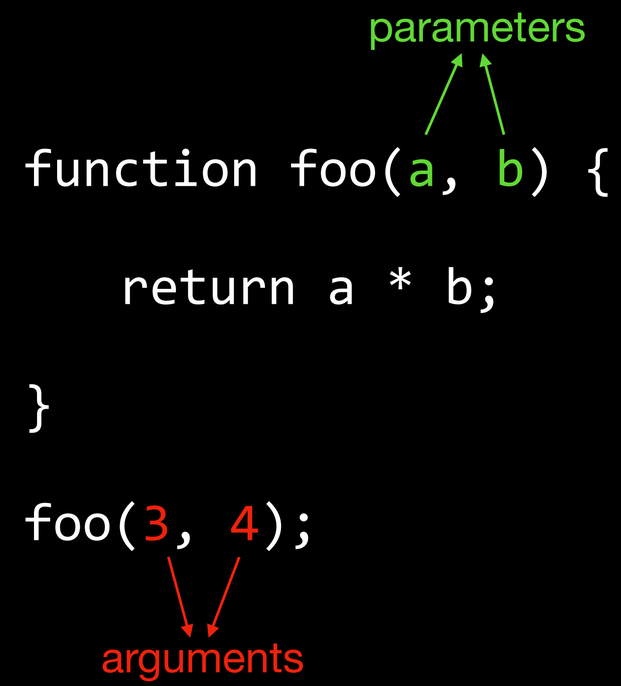
Elements of Programming
Terminology

- A note on ‘arguments’ vs ‘parameters’. (I tend to get confused too!)

```
function foo(a, b) {
    return a * b;
}
foo(3, 4);
```

Elements of Programming
Terminology

- A note on ‘arguments’ vs ‘parameters’. (I tend to get confused too!)



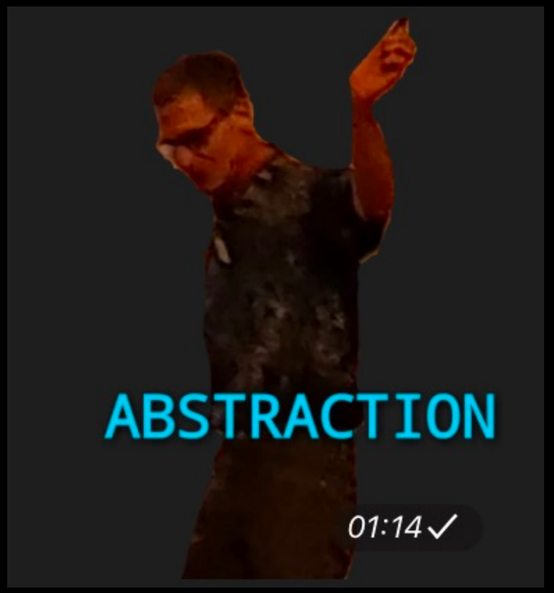
Parameters: names declared in a function declaration.

Arguments: values supplied as part of the function call.

Elements of Programming

Terminology

- ABSTRACTION! (programming is all about abstraction)
  - In a nutshell: to remove complex details (for easier understanding)
  - To apply a function to some arguments:
    - we only need to know WHAT it does
    - but not HOW it does it



End of Recap