COMP10002 – Foundations of Algorithms

Algorithms are fun!
Workshop Time is fun Time!

Workshop Time: 3:20PM – 5:10PM

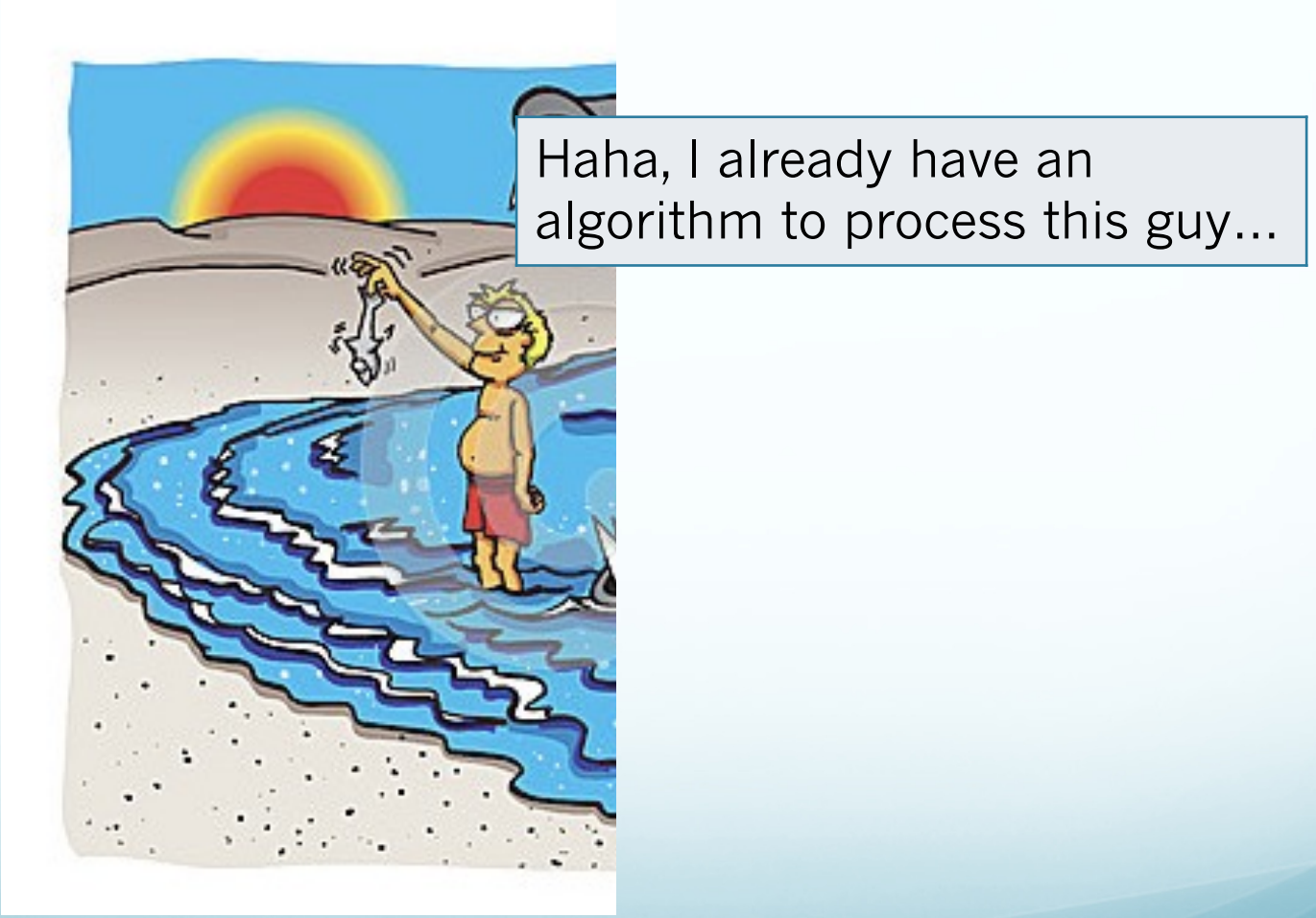
While waiting, please:

- turn on your video,
- say hello to classmates,
- chat & have fun ☺

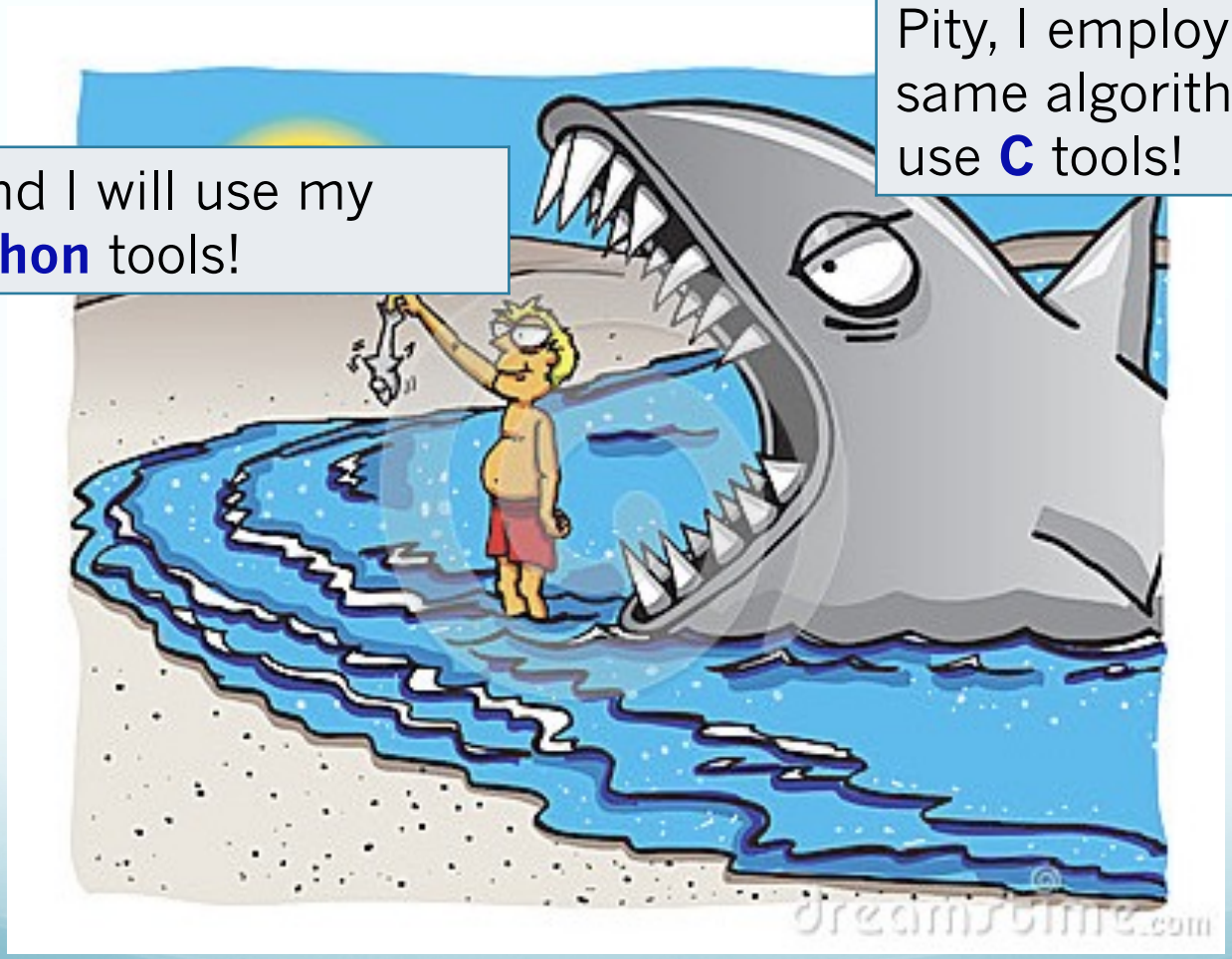
Did you know? You can temporarily unmute your mic by just holding the SPACE key when talking.

About Me: Anh Vo (aka. avo,)

email **avo**@unimelb.edu.au with subject "COMP10002" or just "C102"



Haha, I already have an algorithm to process this guy...


Pity, I employ the same algorithm, but I use **C** tools!
.. and I will use my **Python** tools!

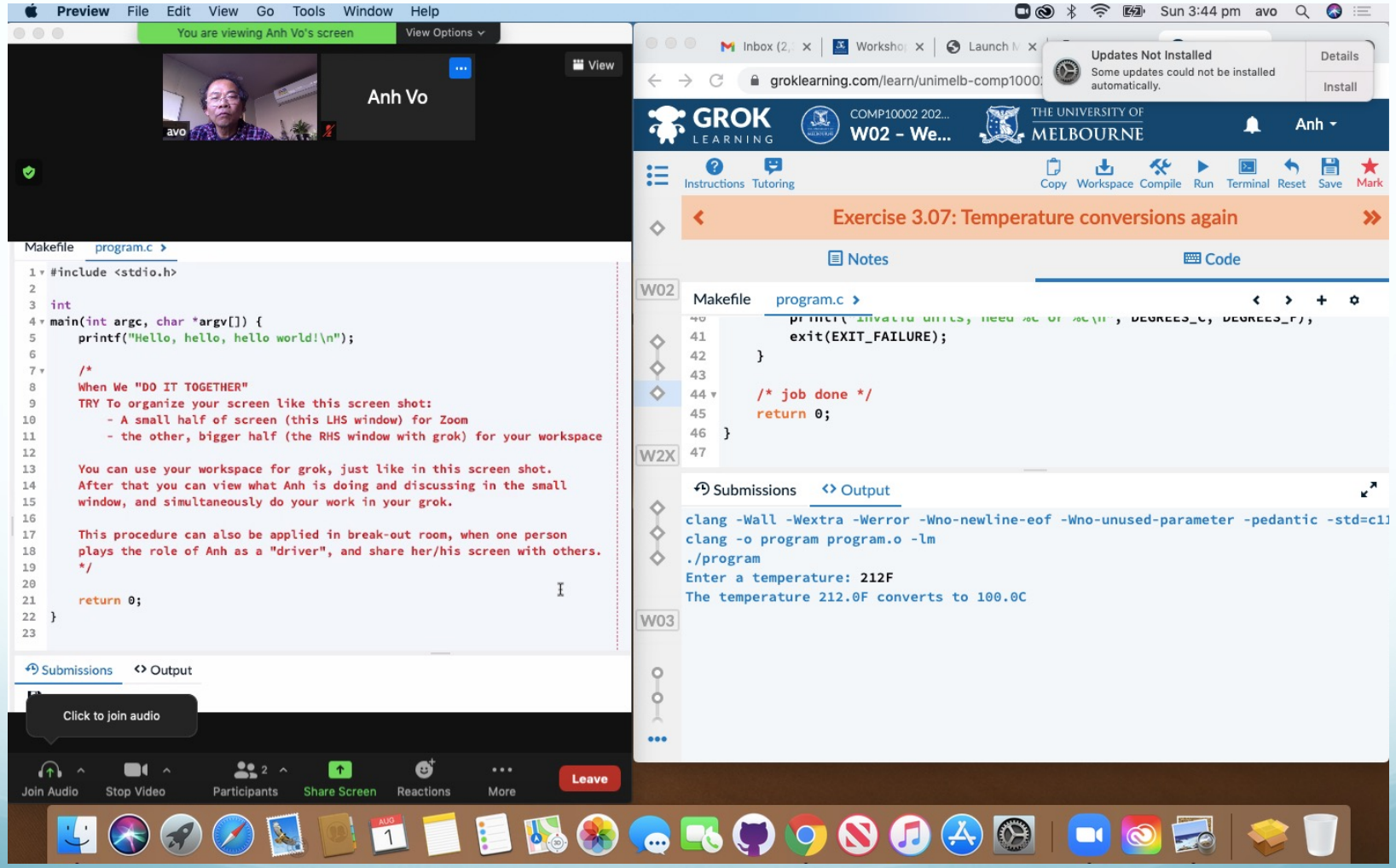
About You – we'll do it a bit later

- In group, have a chat, introduce, get to know some friends
- Tell about your interest, your funny things, your biggest programming/algorithm experience so far

About Us: Workshops

- General:
  - Visit Canvas frequently and explore all contents there
  - Use grok and discussion forum
- Before each workshop:
  - watch all lecture recordings of the previous week
  - try the workshop's exercises (well, as much as you can)
- During a workshop:
  - Be active, start & follow discussions.
  - Turn ON video (if Zooming), have fun, be engaged

Suggestion for arranging your screen during Workshops
Try It Now, together, we'll also visit Anh's github.



Discussion 1: A Problem to Solve

- What steps a computer must do to help us to solve equations ax+b= 0?
- Build an algorithm, and then a C program for that.

program for solving ax+b= 0
Your algorithm: input a, b x= -b/a output x
Anh-the-computer: a 10 b 20 x -2 say “x is -2”

C program: `equation.c`

Opening	`#include <stdio.h>` `int main (int argc, char *agrv[]) {`

Declaring
Inputting
 	
Computing
Outputting
	`float a, b, x;`
	`printf ("a, b = ");`
	`scanf("%lf%lf", &a, &b);`
	`if ( a != 0 ) {`
	`x= -b/a;`
	`printf("Solution x= %f\n", x);`
	`} else {`
	`printf("a must be non-zero\n");`
	`}`

Closing	`return 0;` `}`

Compare with a (imaginable) Python version!

Full C program: `equation.c`

Documentation	`/* Solving equation ax + b = 0`
	`Author: Anh Vo – anhvir@gmail.com`
	`Last updated: 01 Aug 2020 */`

Opening	`#include <stdio.h>` `int main (int argc, char *agrv[]) {`

Declaring
Inputting
Computing
Outputting
	`double a, b, x;`
	`...`
	`/* comments for complicated code segments */`
	`...`

Closing	`return 0;` `}`

Why documentation and indentation? Programs are not just for computers to execute, but also for people to read, understand, and make changes.

Hints: keep your Python indentation habit, but also learn C syntax!

Discussion 2: transition Python → C

- Variables are typed and must be declared before used

  (also note ;)

Python	C

a= 10
b= 5
	int a, b;
	a= 10;
	b= 5;

transition Python → C

- Code block defined by { … }, not by indentation

Python	C

a= 10
b= 5
if (a>b):
  print(“max= ”, a)
  # a is the winner
else:
  max= b
	int a, b;
	a= 10;
	b= 5;
	if (a>b) {
	  printf(“max= %d\n”, a);
	  // a is the winner
	} else {
	  x= b;
	}

transition Python → C

- Input, output using formats: `printf`, `scanf`
- In `printf`, at the end of the format string, add `\n` for end_of_line

Python	C

`a= 10`
`b= 5`
`if (a>b):`
`  print("max= ", a)`
`  print("Done!)`
`else:`
`  ...`
	`int a, b;`
	`a= 10;`
	`b= 5;`
	`if (a>b) {`
	`  printf("%d max= \n", a);`
	`  printf("Done!\n");`
	`} else {`
	`  ...`
	`}`

	Output of this C snippit:
	`10m max=`
	`Done!`

`scanf` is a bit picky!

- copy `my_scanf.c` from `github.com/anhvir/c102`
- paste it to your Editor's window and try it
- we're doing that together right now

Quiz 1

If we execute the following fragment:

```
int i=8;  float x=9;
scanf("%d%f", &i, &x);
printf("i= %d, x= %.1f\n", i, x);
```

with the input stream (data from keyboard) of:

```
  100.0  3.43
```

The output is:

A: `i= 100, x= 3.4`	**B:** `i= 8, x= 9.0`
C: `i= 100, x= 0.0`	**D:** `i= 100 x= 3.43`

Quiz 2

If we execute the following fragment:

```
int i=8;  float x=9; char c= 'A';
scanf("%d%c%f", &i, &c, &x);
printf("i= %d, c= %c, x= %.1f\n", i, c, x);
```

with the input stream (data from keyboard) of:

```
  100AB 3.4
```

The output is:

A: `i= 100, c= AB, x= 3.4`	**B:** `i= 100, c= A, x= 9.0`
C: `i= 100, c=A, x= 3.4`	**D:** `i= 100, c= A, x= 3.4`

Quiz 3

If we execute the following fragment:

```
int i;  char c; float x;
scanf("%d%c%f", &i, &c, &x);
```

with the input stream (data from keyboard) of:

```
100.1A200.2
```

Then, the value of `i`, `c`, *and* `x` *become respectively:*

A: `100  A  200.2`	**B:** `100.1  A  200.2`
C: `100  .  1`	**D:** `(something else)`

Quiz 4: scanf is actually a function, returning the number of data items it *successfully read*

If we execute the following fragment:

```
int i;  char c; float x; int n1, n2;
n1= scanf("%f%d", &x, &i);
n2= scanf("%c%d", &c, &i);
printf("n1= %d, n2= %d, i= %d\n", n1, n2, i);
```

with the input stream (data from keyboard) of:

```
  100.1A200.2
```

The output is:

A: `n1= 1, n2= 2, i= 200`	**B:** `n1= 2, n2= 2, i= 200`
C: `n1= 2, n2= 2, i= 2`	**D:** `(something else or errors)`

C Workspace for Professionals

- You need to have some software installed:
  - A text editor such as `jEdit`
  - A unix-style terminal such as Mac's `Terminal`, or Windows' `minGW`, which includes a C compiler such as `gcc`
- Then, you need to organize a neat working environment, including:
  - Make a dedicated directory (ie file folder) `comp10002`, and sub-folders `Week02`, `Week03`, … under it
  - Arrange the app windows in your screen neatly so that you can view both editor and Terminal windows at the same time, and also can easily switch to your browser when needed.

How to run my equation program on a computer?

1. Type, edit, and save it as `equation.c` using your text editor.
2. Using Terminal and:
   - Run `gcc` to compile `equation.c` to have an executable file with a name such as `equation` or `equation.exe`:

     ```
     gcc –Wall –o equation equation.c
     ```

   - Run the executable file

     ```
     ./equation
     ```

3. If `equation.c` has some errors, go back to step 1

Fun Time

- go to `github.com/anhvir/c102`
- Click on `guessNumber.c` → Raw
- Press ^A^C to copy the Raw content
- Paste it into `grok's` playGround or your Editor Windows
- Try it

Make Lab Time both Useful and Fun

- The exercises are in `grok`, but sometimes incomplete
- You can use `jEdit+Terminal` or `grok`
- Work in breakout rooms (suggestions):
  - One Driver, sharing the screen and do the typing etc.
  - Other people discuss and advise the Driver on what to do while also doing the job in their own computer
  - Change the Driver role after finishing a task, or 15 minutes
  - Call Anh if having problems/questions
- If using grok, you can ask me question, remember to start your question with “avo, “ to get my attention, AND in Zoom call me for help

Lab Time, Play Time ☺

Minimal

Exercise	Where to Find, and Notes
2.08	grok.W02
3.07	grok.W02, but ***incomplete***. It should be read as: *Extend 2.8 program for additional units. Use F for Fahrenheit, C – Celsius, M – miles, K – kilometers, P – pounds, and G for kilograms. (1M= 1.609K; 1P = 0.454G).* *For example:* **H:>**converter **Enter a quantity:** 100M **The distance 100.0 miles converts to 160.9 kilometers.**

Advanced

3.04	grok.W2X: print out the date of tomorrow. Excellent real-life problem, you will use heap of if
NA	Play with guessNumber.c (github.com/anhvir/c102). Try it, explore to understand it, modify it to suit your need and/or make it better.

Wrap-Up

- Algorithms are fun! Stay active, stay happy, stay safe!
- Be engaged: talk, ask friends, Anh, Ed Forum and `Google`

- Workspaces: `Canvas`, `grok`, ready with `JEdit`+`Terminal`
- Programs: structure, editing, compiling, running, testing
- C - Python: typed variables, `printf`, `scanf`, syntax, and … faster!
- But similar: algorithmic thinking, and indentation ☺ .

type	`int`	`float`	`double`	`char`	*`string`*
printf format	`%d`	`%f`	`%lf`	`%c`	`%s`
scanf format	`%d`	`%f`	`%lf`	`%c`	`%s`
scanf for `v`	`&v`	`&v`	`&v`	`&v`	`v`

See You Next Week!

It’s time, and you can leave.

But if you still have questions, you can stay with me for a few more minutes.

Lab implementation: Ex. 2.8 & 3.7

***2.8**: To convert from degrees Fahrenheit to degrees Celsius, you must first subtract 32, then multiply by 5/9. Write a program that undertakes this conversion.*

***3.7**: Extend 2.8 program for additional units. Use F for Fahrenheit, C – Celsius, M – miles, K – kilometers, P – pounds, and G for kilograms. (1M= 1.609K; 1P = 0.454G).*
For example:
H:>converter
Enter a quantity: 100M
The distance 100.0 miles converts to 160.9 kilometers.