COMP20005
Engineering Computation

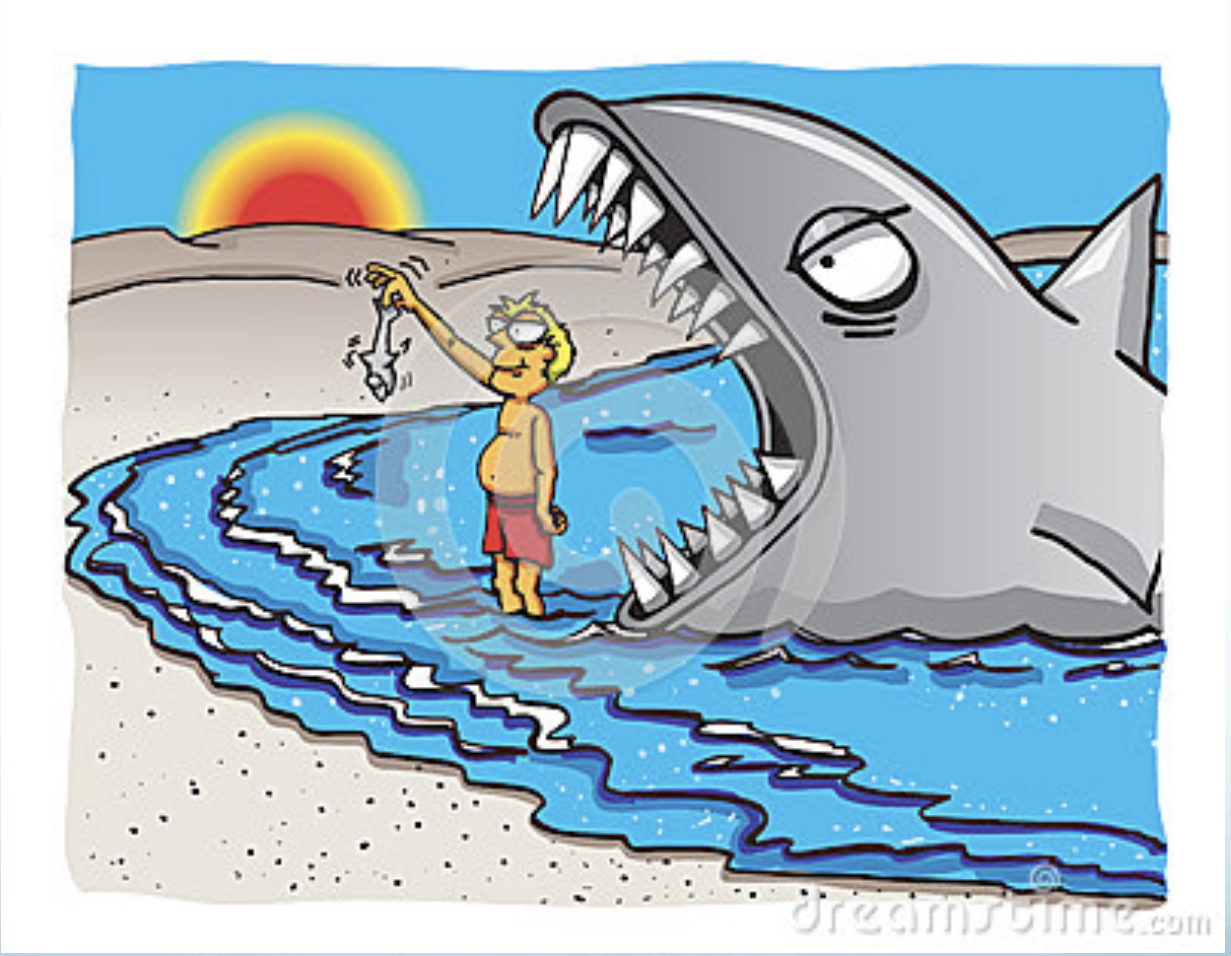
Welcome to the First Workshop!

NOW: spend a few minutes to

- **greet and know friends around you, and**
- **tell them what you expect from the workshops, or just a funny thing about yourself.**

Greetings from Anh Vo

- email avo@unimelb.edu.au , subject starting with “COMP20005” or just “C205”



Administrivia + Expectations

- Workshop:
  - the first half: class discussion, and
  - the second half: individual work on laptops or lab PCs.
- Students:
  - Have pens/pencils and papers ready 🤪🤪
  - Be Active && Work && **Talk** && **Have Fun**!
  - Be prepared for the workshops (at home, please):

    Canvas→Modules→Week 2 (then Week 3 etc.)

    the read the "Workshops" details & prepare
- Tutors: help, be students' peers & friends!

First Problem to Solve

- solve the equation ax+b = 0 ???

First Problem to Solve

- “program” Anh to solve the equation ax+b = 0 for you

First Problem to Solve

- Suppose that I (Anh) am a typical computer, capable of:
  - inputting data (listening),
  - outputting data (speaking), and
  - using my memory to store data, and carrying all kind of arithmetic computations.
- Your talk: “program” me to solve the equation ax+b = 0

Your Solution
(algorithm for Anh-a-computer)

Today's Program

```
1   int main(int argc, char *argv[]) {
2     greet("class mates", 5);
3     understand("C programs");
4     know("equipment and tools");
5     have_fun();
6     implement("ex 1.2");
7     implement("ex 2.8");
8     if ( time_permitted() ) {
9       implement("ex 2.4");
10    }
11    return 0;
12  }
```

Second (Engineering) Problem
build program for computers, not me :-)

Problem:	**Solve equation ax + b = 0**
Program:	

A computer program

Problem:	**Solve equation ax + b = 0**
Program:	`x= -b/a;`

A computer program

Problem:	**Solve equation ax + b = 0**

Program:	`input value of a and b;`
	`x= -b/a;`
	`output value of x;`

Memo:	A typical computer program has 3 sections: 1. inputting *data* 2. computing *results/solutions* 3. outputting *results/solutions*

C program: equation.c

Inputting	`scanf(a);`
	`scanf(b);`
Computing	`x= -b/a;`
Outputting	`printf(x);`

C program: equation.c

Opening	`#include <stdio.h>` `int main (int argc, char *agrv[]) {`
Declaring	`double a, b, x;`
Inputting	`printf ("Enter value of a and b: ");` `scanf("%lf %lf", &a, &b);`
Computing	`x= -b/a;`
Outputting	`printf("Solution x= %lf\n", x);`
Closing	`return 0;` `}`

Full C program: equation.c

Documentation	`/* Solving equation ax + b = 0`
	`Author: Anh Vo – anhvir@gmail.com`
	`Last updated: 07 Mar 2019 */`

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

Declaring
Inputting
Computing
Outputting
	`double a, b, x;`
	`...`
	
	`printf("Solution x= %lf\n", x);`

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.

Program – Computer - Compiler

How to run equation.c on grok

1. Type, change (edit) it in `PlayGround`
2. Click `Run`
3. If (having some errors or warnings) go back to Step 1
4. Click `Mark`

Time for fun

Goto `github.com/anhvir/c205` then:

Click on **`guessNumber.c`** → **`Raw`**

Copy the content **`(Ctrl-A`** then **`Ctrl-C)`**

Paste to **`PlayGround`** of **`grok`**

Try **`Run`**

Try to modify the program, for example by changing **`"Anh"`** to your name, by changing **`MAX`** from **`10`** to **`5`** or something else .

Using github in this class

- For each week, the directory:

  `github.com/anhvir/c205`

  normally has a new content which is useful for this class. Note if you want to keep the material of that `c205` you need to download it to your computer every week.

- In the lab, first you should use a browser to navigate to that website `and` follow the instructions there.

Lab

- Using `grok` to do all exercises
- Try `guessNumber.c`
- Put your hand up if you have questions or if you have something exciting to say

Remember

- Stay active, stay happy!
- Work, Talk, Ask friends, tutors and Mr `Google`
- Use `LMS Canvas`, `grok`, `jEdit`, `minGW`, `github`
- Programs: structure, editing, compiling, running, testing
- Variables: names, data types, values
- Input with `printf` and `scanf`, output with `printf`

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

Ex: 1.2, 2.8 & 2.4

1.2: *Implement* `helloworld.c`

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. For example:

H:>`converter`
Enter a temperature in Fahrenheit scale: `212`
The temperature 212.0F converts to 100.0C

2.4*: Write a program that has* `<limits.h>` *and* `<float.h>` *included at the top, and then print out the values of the following constants:* `INT_MAX, INT_MIN, FLT_MIN, FLT_MAX, DBL_MIN, DBL_MAX`. (these constants are pre-defined in the above .h files)
Do the printed values agree with those in the textbook?
Note: You can use github `c205/e24.c` *as a template.*