COMP20005

Intro to Numerical Computation in C

aka. *Programming is fun*

Welcome to the First Workshop!

When waiting:

- greet and know friends around you, and
- **login into your laptop (or lab's PC if no laptop), then open `LMS` and `grok`.**

Today's Plan:

- How to make programs for computers?
- 5-min break for stretch exercises, networking, and fun
- using grok for exercises 1.2, 2.8. 2.4 and more

suggested WhatToDos for Workshops (Learning-By-Doing)

in your own time

W=1

Attend & Revise lectures of week W

W++

- Prepare according to
  `LMS`→Weekly Schedule→Week W→Workshops
- Practise, practise, practise
- Extensively use `grok`

- Finish the outstanding tasks
- Check with the solutions provided if needed

in workshop

in the first part (tutorial):

- be active in discussions:
  - analyse/find approach/do the problems
  - ask, answer questions, vote
- be cooperative.

5 min for stretch exercises and social networking

in the second part (lab):

- be cooperate & active
- implement in computers, implement as many as possible

directly (or via emails) tell Anh on what else you want in the workshops

Your First Numerical Program

- "program" Anh to solve the equations ax+b = 0 for you
- Suppose that Anh is a typical computer, capable of:
  - inputting data (listening),
  - outputting data (speaking),
  - using my short-term memory to store named data, and
  - using my CPU (brain) to carry all kind of arithmetic computations.
- Your talk: teach (ie. "program") Anh to solve the equations ax+b = 0 for you

Your program to solve ax+b=0 (step-by-step algorithm for Anh-a-computer)

```
Start
    1. do ???


Stop
```

A computer program

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

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

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

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;` `}`

Editing & Compiling Your Codes

- **Method 1** (used in workshops): using `grok`
  - Pros:
    - cloud-based, simple, easy to use, excellent for most of workshop exercises ☺
    - providing convenience for instructors to support students
    - safe: your programs won't be lost!
  - Cons:
    - limited ability in programming, you won't learn much.

- **Method 2**: using `Visual Studio Code` + `gcc`
  - Pros: offline, powerful, helps to understand more, especially useful for assignments and big programs.
  - Notes:
    - this week: install `VSC` and `gcc` in your laptop in your own time (see `LMS`)
    - we will introduce next week.

Method 1 Together: 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

```
DO TOGETHER WITH Anh
Go to github.com/anhvir/c205
Copy the text content of equation.c
Paste it on grok's PlayGround
Try, and make it working
```

Full C program: equation.c

Documentation	`/* Solving equation ax + b = 0`
	`Author: Anh Vo – avo@unimelb.edu.au`
	`Last updated: 07 Mar 2022 */`

Opening	`#include <stdio.h>` `int main (int argc, char *agrv[]) {`
Declaring	`double a, b, x;`

Inputting	`// inputs a and b`
	`printf ("Enter value of a and b: ");`
	`scanf("%lf %lf", &a, &b);`

Computing	`// computes x as solution to ax+b= 0` `x= -b/a;`
Outputting	`// outputs result` `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.

Today's Program

```
1   int main(int argc, char *argv[]) {
2     understand("workshop format");
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  }
```

5-minute break

- stress exercises
- social networking

Time for fun

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

Click on **guessNumber.c**, then **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 by Friday every week.

Lab

- Using `grok` to do all exercises of Week 2
- (if not yet done) Try `guessNumber.c` (downloaded from `github.com/anhvir/c205`)

- Help your mates, and/or ask your mates for help. Make noise!
- When doing a grok exercise, ask Anh for online support by click on 

- Put your hand up to:
  - give Anh questions
  - tell Anh that you discover something funny, or exciting, or extraordinary

Important Homework

1. If you haven't installed `Visual Studio Code/jEdit` on your laptop, do it at home ASAP and within this week. Instruction for installation is available in **LMS**: `LMS` --> `Modules` --> `Working With Grok and ...` --> `Install gcc and Visual Studio on Your Own Computer`
2. Remember: `grok` is a web interface, and you cannot use it offline. In addition, `grok` probably does not support full functionality of a programming environment. As a professional, you'd better to also have `VSC/gcc`. Install them today!

Remember

- Stay safe, stay active, stay happy!
- Use `LMS`, `grok`, `VSC`, `gcc`, `github`

- Variables: names, data types, values
- Input with `printf` and `scanf`, output with `printf`
- Data types and respective formats for `printf`, `scanf`:

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

Programming is fun!