

COMP10002 Workshop Week 2

Welcome to the First Workshop!

```
0 #include ...
1 int main(int argc, char *argv[ ]) {
2     do("greeting");
3     understand("algorithms and programs");
4     learn("equipment and tools");
5     have_fun();
6     implement("ex 2.8");
7     implement("ex 3.7");
8     if (time_permitted) {
9         explore("guessNumber.c");
10    }
11    return 0;
12 }
```

*Please
Login into
lab's computer
rightnow!*

First Problem

- I am a typical computer (well, sort of)!
- Someone (say, Alistair) wants to get some information about the class (such as students' name, major, year, programming experience...), and also wants each of you get to know some (say 2-5) classmates.
- Your talk: build algorithm (program) for me to do the above task!

Algorithm

- Need a correct, and efficient algorithm!

Administrivia + Expectations

- Each Workshop:
 - first hour: tute & class/group discussion, and
 - second hour: individual work on lab PCs or laptops.
- Students:
 - Be prepared for the workshops.
 - Be active in the class, start & follow discussions.
 - Work, Talk, AND practice soft skills!
 - Use LMS, also use github.com/anhvir/c102

Second 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 <i>data</i> 2. computing <i>results</i> 3. outputting <i>results</i>

C program: equation.c

Inputting

```
scanf(a);      ???
```

```
scanf(b);      ???
```

```
x= -b/a;
```

```
printf(x);    ???
```

**Computing
Outputting**

C program: equation.c

Opening	#include <stdio.h> int main (int argc, char *argv[]) {
Declaring Inputting	float a, b, x; printf ("a, b = "); scanf("%f%f", &a, &b); if (a != 0) { x= -b/a; printf("Solution x= %f\n", x); } else { printf("a must be non-zero\n"); }
Computing Outputting	
Closing	return 0; }

Full C program: equation.c

Documentation	<pre>/* Solving equation ax + b = 0 Author: Anh Vo - anhvirl@gmail.com Last updated: 07 Mar 2017 */</pre>
Opening	<pre>#include <stdio.h> int main (int argc, char *argv[]) {</pre>
Declaring Inputting Computing Outputting	<pre> float a, b, x; ...</pre>
Closing	<pre> return 0; }</pre>

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

Remember

- Simple data type include **int**, **float**, **double**, **char**
- **printf** and **scanf** need a bit of remembrance
- Simple rules:

type	int	float	double	char	<i>string</i>
printf format	%d	%f	%f	%c	%s
scanf format	%d	%f	%lf	%c	%s
scanf for v	&v	&v	&v	&v	v

Your uni's directory H:

- The directory is accessible from anywhere (if you do things properly). Make sure that you can copy files between your laptops and `dimefox/nutmeg` no matter where you are.
- It's better:
 - to have a directory `comp10002`, and
 - directories `week2`, `week3`, ... and
 - separate directories for assignments.

Now try lab programming environment

- The task:

- Start `jEdit` and `minGW`
- Build your directory system in `H:.`. In `minGW` window do

Command	Explanation
<code>cd H:</code>	<code>cd</code> = change working directory
<code>mkdir comp10002</code>	make a directory for this subject
<code>cd comp10002</code>	make <code>comp10002</code> the working directory
<code>ls</code>	display content of the (working) directory (result should be empty)
<code>mkdir week2</code>	make a new sub-directory under <code>comp10002</code>
<code>cd week2</code>	make <code>week2</code> the working directory

Using lab computers & github

- For some weeks, the directory `github.com/anhvir/c102` normally has a new content which is useful for this class.
- Now, navigate to that website and download the whole directory `c102` to your computer, then:
 - copy the whole directory to `comp10002/week2/`
 - use `jEdit` to open and fix `equation.c`,
 - use `minGW` window to compile and run `equation.c`

Programming environment at home

- If you have a MacBook:
 - install [jEdit](#)
 - install [gcc](#)
 - use [Terminal](#) for running [gcc](#) and testing programs
- If you have a Windows laptop:
 - install [jEdit](#)
 - install [minGW](#) (remember to also mark [openssh](#))
 - use [minGW](#) for running [gcc](#) and testing programs
- How to install? Read instruction in [LMS](#).
- For working from home:
 - should install [VPN](#),
 - learn to use [scp](#) and [ssh](#)

Time for fun

If not done, download github.com/anhvir/c102 then:

Compile, run and explore `guessNumber.c` using:

```
cd c102
```

```
gcc -Wall -o guessNumber guessNumber.c
```

```
guessNumber
```

Later: explore `guessNumber.c`, and

- learn about `string` data type, `while` and `rand()`.

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.

Remember

- Programming is fun!
- Stay active, stay happy!
- Program, talk, ask friends, tutors and Mr Google
- Use LMS, jEdit, minGW, Chrome, github
- Programs: structure, editing, compiling, running, testing
- Variables: names, data types, values
- `if (?) {...} else {...}`
- Input with `scanf`, output with `printf`:
- Data types and formats for `printf`, `scanf`:

type	<code>int</code>	<code>float</code>	<code>double</code>	<code>char</code>	<code>string</code>
<code>printf</code> format	<code>%d</code>	<code>%f</code>	<code>%f</code>	<code>%c</code>	<code>%s</code>
<code>scanf</code> format	<code>%d</code>	<code>%f</code>	<code>%lf</code>	<code>%c</code>	<code>%s</code>
<code>scanf</code> for <code>v</code>	<code>&v</code>	<code>&v</code>	<code>&v</code>	<code>&v</code>	<code>v</code>