

# COMP1038

# AE1PGA

## PROGRAMMING AND ALGORITHMS

2024 — 2025

Jiawei Li (Michael)

# MOODLE

The primary source of information for the course is [Moodle](#). I will link to any information rather than copying it to avoid conflicts if things change.

This module is also known as **COMP1038**.

# CONVENERS



Jiawei Li (Michael)

PMB426



Pushpendu Kar

PMB448



Heshan Du

PMB448

# TEXTBOOK

## C: How to Program (8th ed)

We don't have time to go through the entire book.

Chapters 1-14 are relevant for this module.

Reading technical documentation is a skill you will pick up as you go through your degree.

# ASSESSMENT

## 75% COURSEWORK

- 30% CS1 programming assignment
- 45% CS2 programming assignment

## 25% EXAM

- 1 hour

Don't panic.

# EXPECTED MARK DISTRIBUTION

1 <sup>st</sup> (marks 70 or higher)	34.9% (61)
II-1 (marks 60 to 70)	27.4% (48)
II-2 (marks 50 to 60)	13.1% (23)
3 <sup>rd</sup> (marks 40 to 50)	17.1% (30)
Fail (marks less than 40)	7.4% (13)
Number of students	175
Number of borderline marks	12
Number of non-submissions	4

As guidance, QSC expectations are that around **30%** of students gain a First Class degree.

# GOAL

- Programming language concepts
- Programming in C
- Role of algorithms in programming

# TOPICS (I)

- Programming language concepts overview
- Programming language concepts in C
- Structured programming (instructions + data)
- Program Control (instructions)
- Data (types + variables)
- Functions
- Arrays
- Pointers
- Memory management
- Strings
- I/O + Files



# TOPICS (II)

How to:

- analyse a problem statement
- design a program
- test a program is correct
- debug an incorrect program

# TOPICS (III)

How to:

- use a compiler, command line, and other programming tools
- use a text editor designed for programming
- work in a Unix (Linux) environment
- read developers documentation
- and a collection of other "flexible" skills

# RESOURCES

You will not be given every detail in lectures.  
Consult your notes, textbook, ask in labs, etc.

(this implies you need to take notes)

You will be expected to check your question doesn't  
have a obvious answer before asking.

# LANGUAGE

## C

**C** is a statically typed, structured, general purpose computer programming language which has formed the basis for many other programming languages.

# Keywords in C

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	static	void
default	goto	sizeof	volatile
do	if	signed	while

# FEEDBACK

- Forum questions
- Lab questions
- Unassessed assignments
- Assessed assignments

Email questions will be reposted to the forum, so please just post them there to begin with.

# WHAT IS A PROGRAM?

An organized list of instructions that, when executed, causes the computer to behave in a predetermined manner. Without programs, computers are useless.

# CODE + DATA

A program can be seen as code + data. It contains a list of ingredients (called **variables**) and a list of directions (called **statements**) that tell the computer what to do with the variables. The variables can represent numeric data, text, or graphical images



# INPUT & OUTPUT

Input/Output (I/O) describes any operation, program, or device that transfers data to or from a computer. Typical I/O devices are printers, hard disks, keyboards, and mice.

# INTERACTION?

- Interactive programs

The user enters some or all of the data into the program while it is running.

- Batch processing

The user does not enter any data directly into the program. All data required is taken from files, network, etc.

- Command line arguments

Provide extra data to the program at the time you start running it. The program may then proceed to run as an interactive or batch program.

# WHAT IS A PROGRAMMING LANGUAGE?

A way of explaining to the computer what we want it to do.  
Allows us to express our needs at a level that is easy for us  
(humans) to understand.

Different programming languages are designed to make  
different things easy to express.  
(which implies not all programming languages are good at  
every task)

Good languages make a wide range of tasks easy.

**HIGH-LEVEL**

JAVA, Python, ...

vs

C is down here

**LOW-LEVEL**

Machine language isn't far below

# PROGRAMMING LANGUAGE CONCEPTS

Programming can be researched with abstract, mathematical models (Turing machines).

Programming concepts exist independently from languages that implement them.

For comparison, the concept of verbs  
vs.

verbs in English, verbs in Spanish, verbs in Mandarin.

# PROGRAMMING LANGUAGE CONCEPTS

We will talk about programming language concepts directly in the first few hours.

The rest of the course will show which concepts are implemented in C, and how to use them.

In other modules you will see which concepts are implemented in other programming languages.

# X2Go

This is what we use to edit, compile, and execute our programs (in this semester).

# PROCESS OF C PROGRAMMING

1. Write your source code in a text editor.
2. Compile your source code with [gcc](#).
3. If there is a compile-time error, go to 1 and fix it.
4. If not, run your newly compiled executable.
5. If it does not work correctly, go to 1 and fix it.



# HELLO WORLD in C

- Any header files which define needed external functions.
- All C programs require a main function.
- Program starts at the beginning of main.
- Program ends at the end of main.