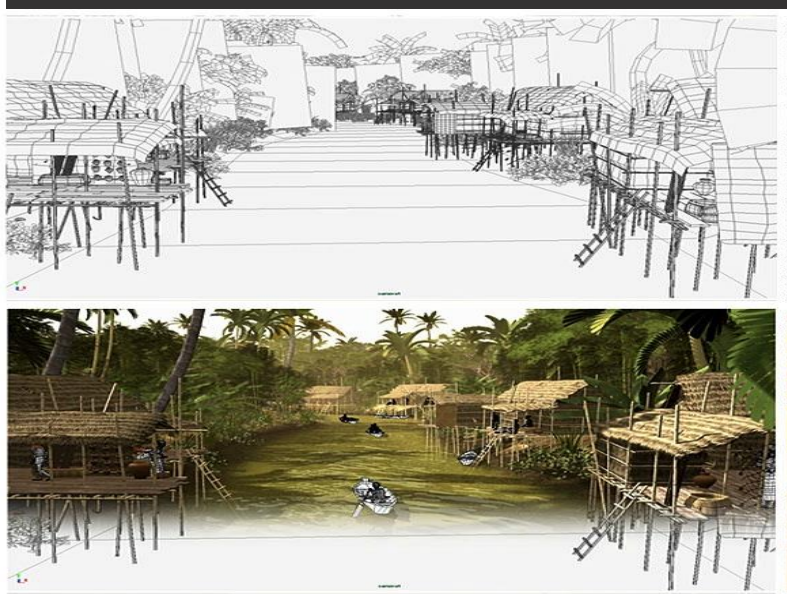
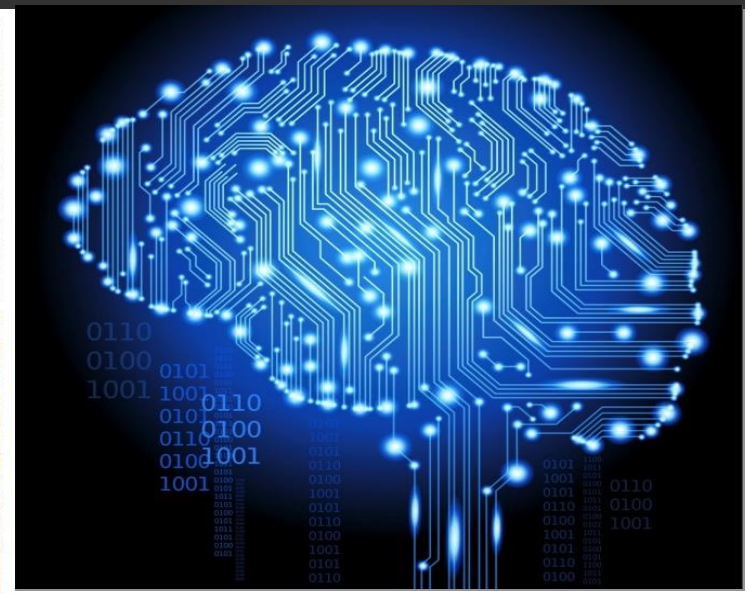
MONASH University

Information Technology

FIT1008/2085
Introduction – Unit Basics

Prepared by:
Maria Garcia de la Banda
Revised by A. Aleti, D. Albrecht, G. Farr, J. Garcia and P. Abramson





Learning Objectives

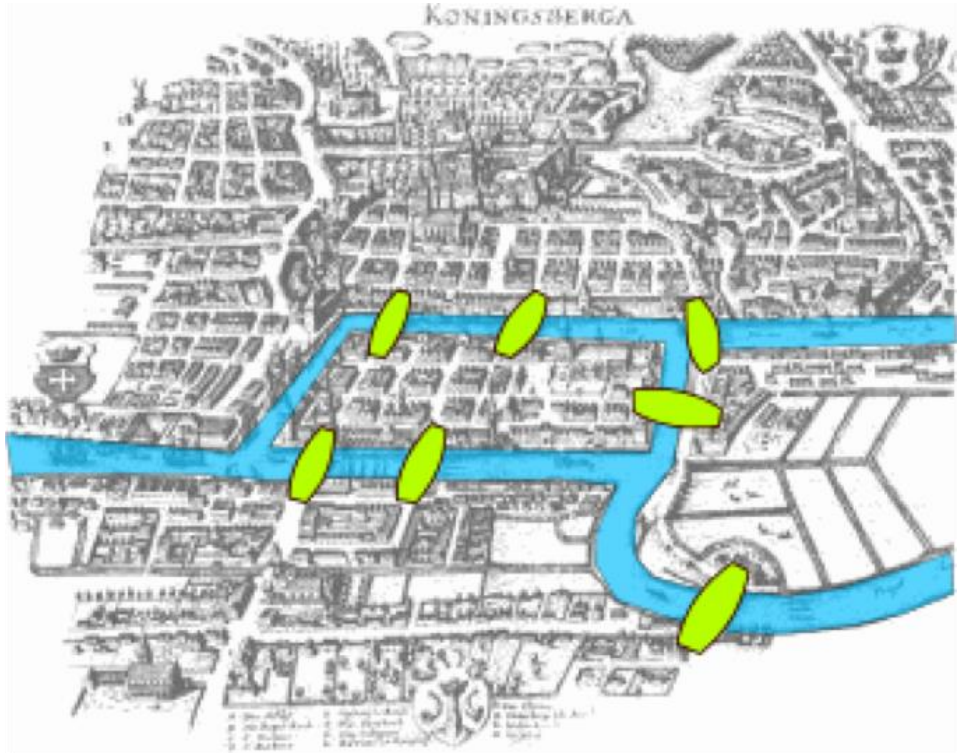
- **To learn about the unit basics:**
  - The teaching team
  - What it is(are) the unit(s) all about
  - What will you get from this unit
  - Main differences between FIT1008 and FIT2085
  - Weekly topics
  - Recommended reading

What is the unit all about?

What is this unit all about?

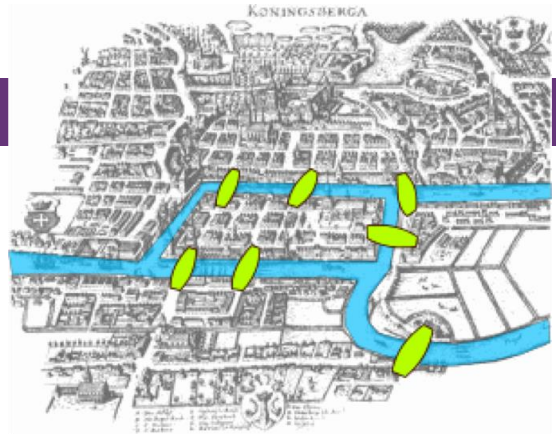
- **It is about the fundamentals of Computer Science and Software Engineering**
- **It all starts with a problem that needs to be solved**

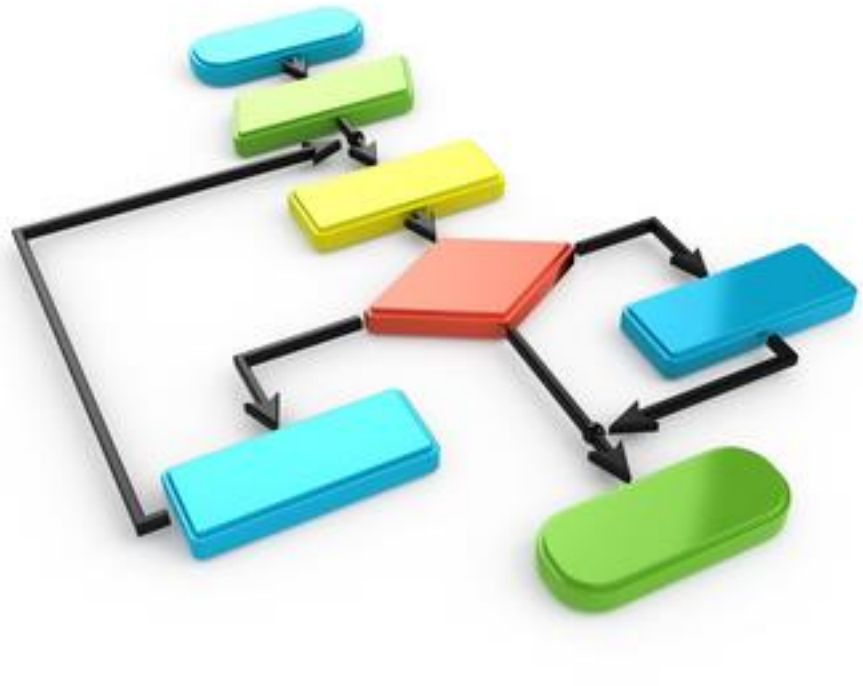
- **Consider the famous Königsberg seven bridges problem:**
  - Find a walk through the city that crosses each bridge once, and only once



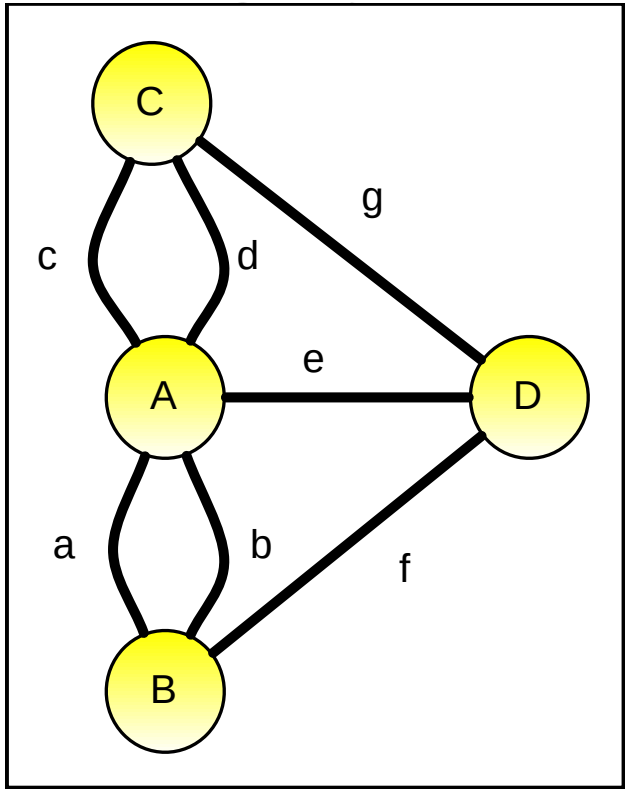
What is this unit all about?

- To do this we often abstract it
- & develop a high-level algorithm





V = [A, B, C, D]



What is this unit all about?

- **We then encode the algorithm in a programming language**

```
def swap(the_list, i, j):
    the_list[i], the_list[j] = the_list[j], the_list[i]


def selection_sort(the_list):
    n = len(the_list)
    for k in range(n):
        min_position = find_minimum(the_list, k)
        swap(the_list, k, min_position)


def find_minimum(the_list, starting_index):
    min_position = starting_index
    n = len(the_list)
    for i in range(starting_index, n):
        if the_list[i] < the_list[min_position]:
            min_position = i
    return min_position
```

What is this unit all about?

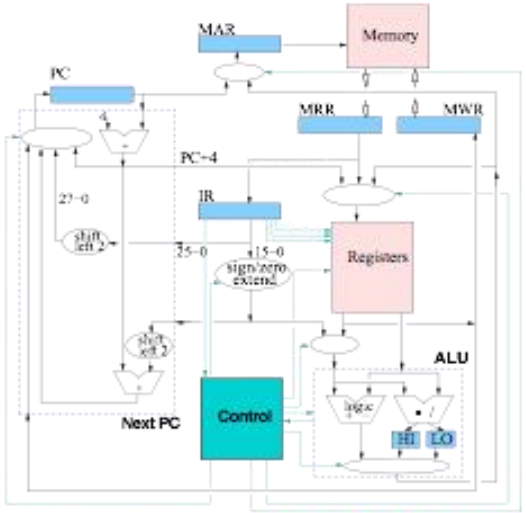
- **Which is compiled/interpreted into assembly language**

```
1                   .data
2   A:              .word 10        # change value to desired number for A, which is a0
3   B:              .word 4         # change value to desired number for B, which is a1
4   array:          .word 0:50      # size must be changed to accommodate A snd B
5                   .text
6
7   main:
8           lw $s0, A               # $s0 = A
9           addi $s0, $s0, -1       # to accommodate for loop condition
10          lw $s1, B               # $s1 = B lowercase b turns blue... WHY
11          addi $s1, $s1, -1       # see line 9
12          la $s2, array           # "look at" address of array
13          li $s3, 0               # set i = 0
14          li $s4, 0               # set j = 0
15
16  For1:
17          blt $s0, $s3, Exit      # for(i = 0; i < A; i++)
18          addi $s3,$s3,1          # i++
19          li $s4, 0               # resets j to 0 after each iteration of the for loop
20          j For2                  # executes the nested for loop
```

What is this unit all about?

- **Then it is assembled and linked into machine language, and executed in a CPU**

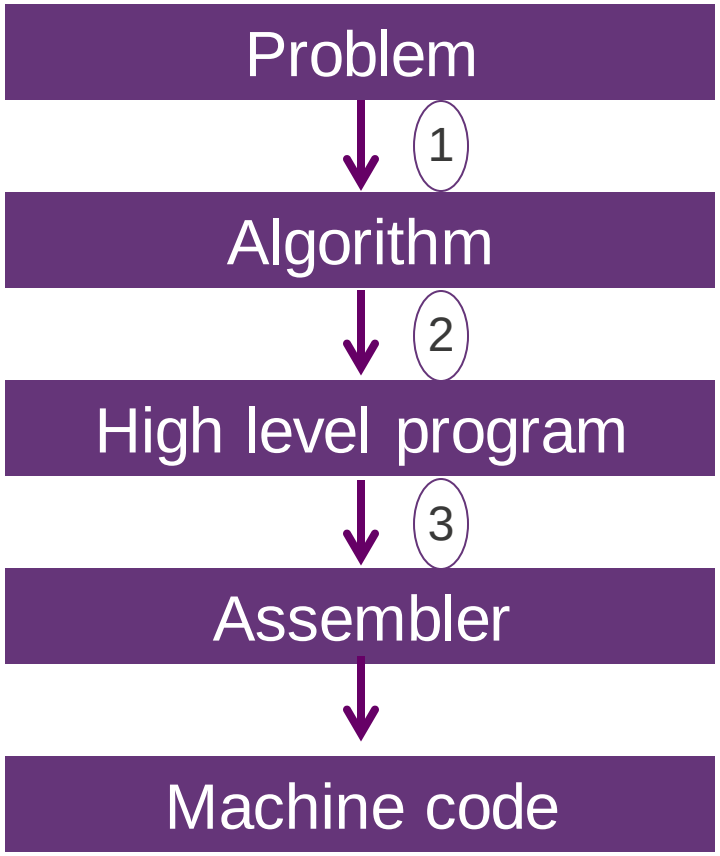
Machine language



CPU

What is this unit all about?

- **We will focus on basic knowledge that helps with the first three steps**

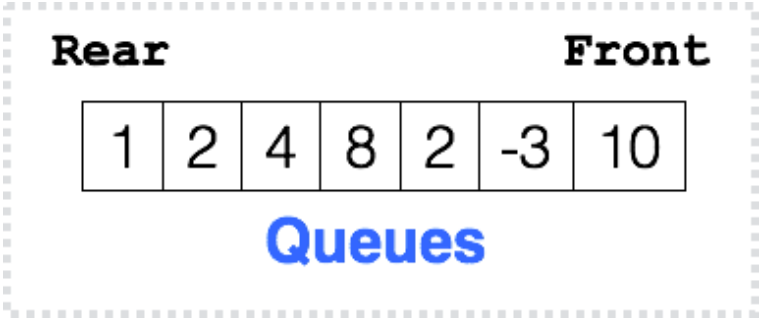


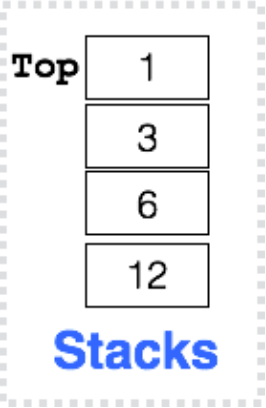
What you will get from this unit?

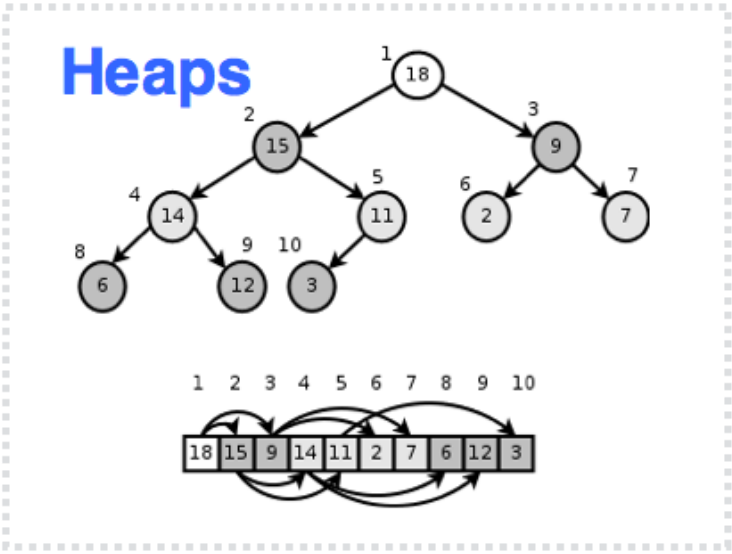
What you'll get from this unit

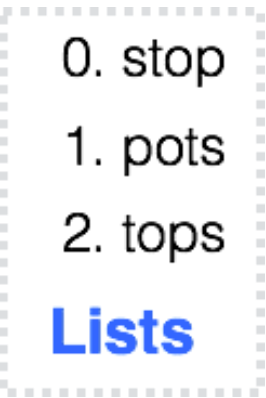
- **Implement and modify many different data types**

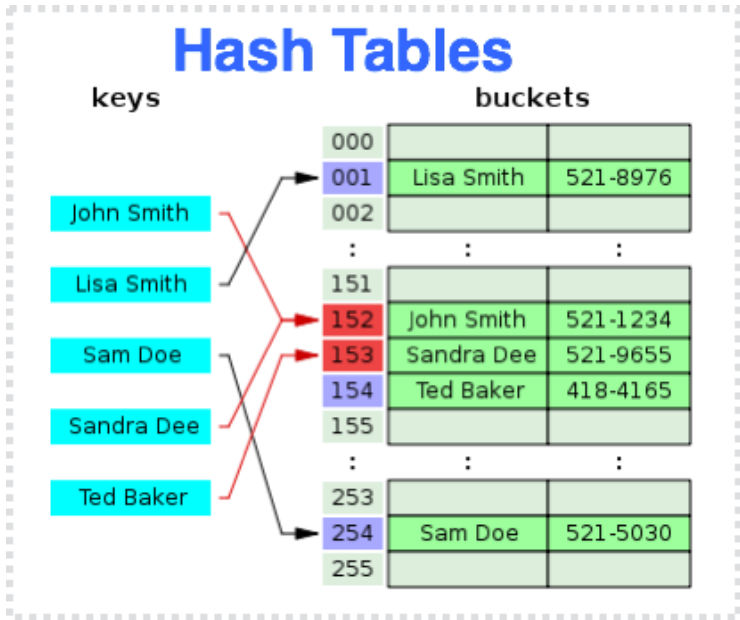
Some of the Data Types you will see











What you'll get from this unit

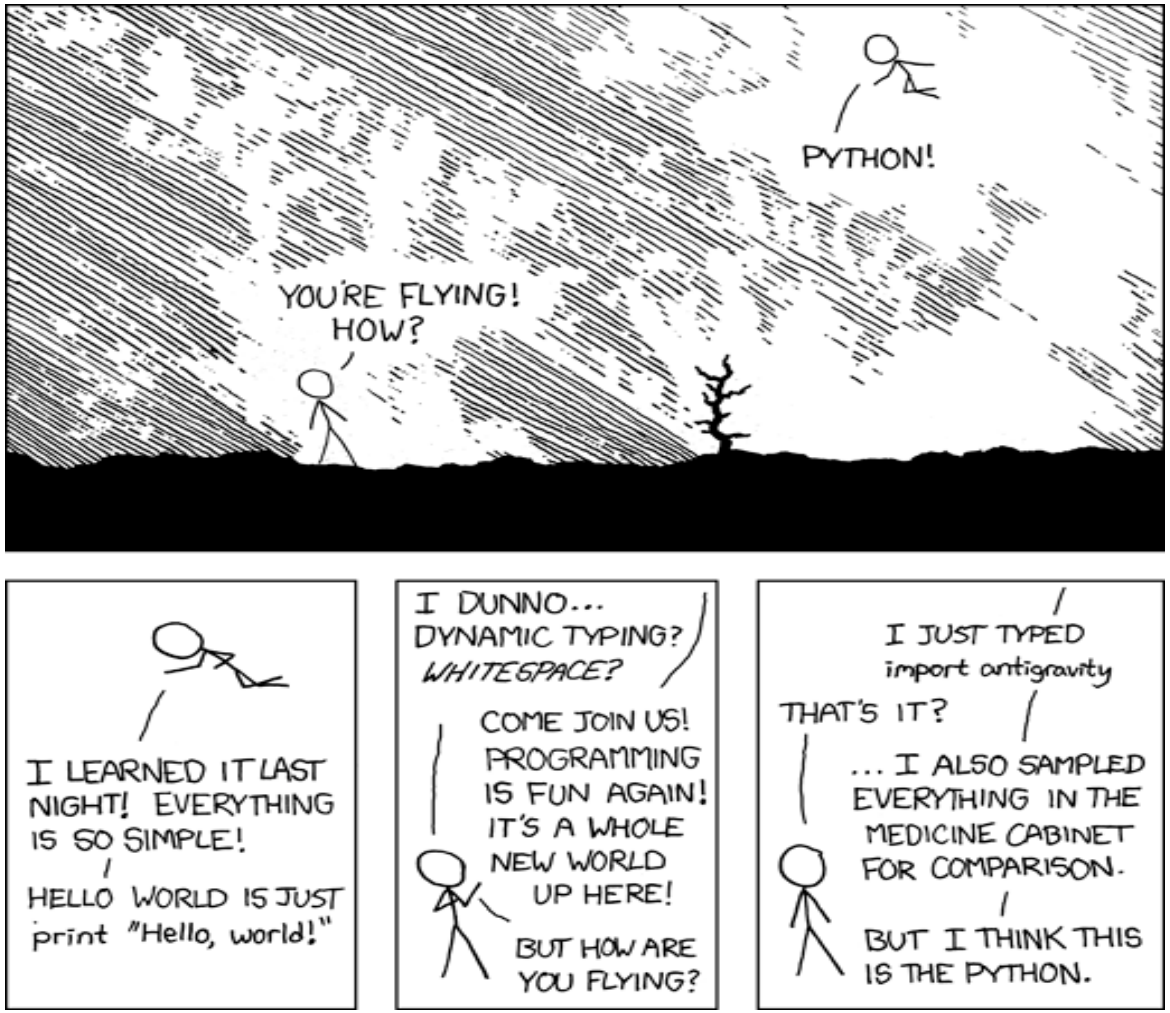
- Implement and modify many different data types
- Evaluate/compare different data type implementations
- Design, implement and test algorithms on data types

Programming Language to implement

- **We will use Python (3.6+) not JavaScript**
- **Why?**
  - General purpose (not web focused)
  - Simple syntax, so better to learn basics
  - Has depth (mutli-paradigm)
  - Great libraries
  - Also very popular

Important for type-hints.
3.8 recommended

xkcd.com



Important!

- **This course is about learning/practicing the CS/SE fundamentals**

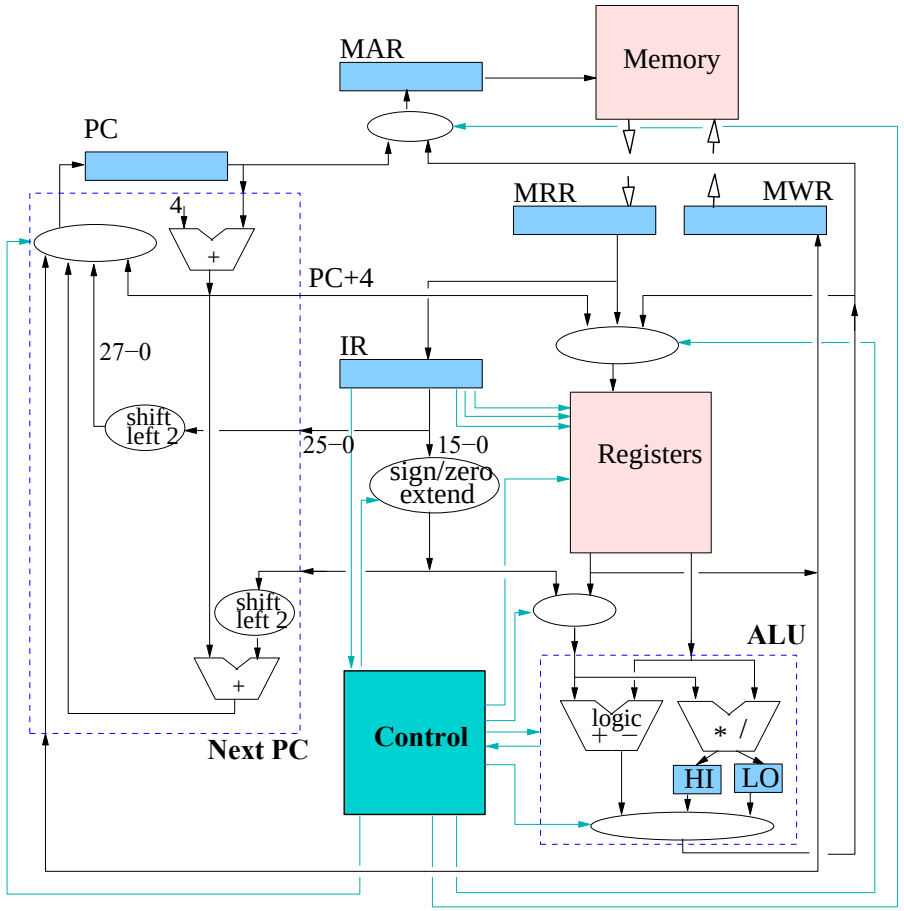
This is NOT a Python course

- **Python is only used as a tool to illustrate the concepts**
- **Sometimes we will have to bend it a bit…**
- **DISCLAIMER: I am not a Python programmer expert!**
  - Not necessary for this unit

What you'll get from this unit

- **Implement and modify many different data types**
- **Evaluate/compare different data type implementations**
- **Design, implement, document and test algorithms on data types**
- **How simple high-level programs are compiled into MIPS assembler**
  - Look "under the hood" of high-level programming languages
  - How your code runs

MIPS Architecture & assembly language




```
        .text
fact:   ori     $v0, $0, 1
        addi    $s0, $0, 1

loop:   slt     $t1, $s0, $t0

        bne     $t1, $s0, end

        mult    $v0, $t0
        mflo    $v0

        addi    $t0, $t0, -1

        j       loop

end:    jr      $ra
```

What you'll get from this unit

- **Implement and modify many different data types**
- **Evaluate/compare different data type implementations**
- **Design, implement, document and test algorithms on data types**
- **How simple high-level programs are compiled into MIPS assembler**
  - Look "under the hood" of high-level programming languages
  - How your code really works

Main aim: learn and/or practice the fundamentals

Main differences FIT1008/FIT2085

Main differences among FIT1008/2085

- **FIT1008 belongs to the Computer Science degree**
- **FIT2085 belongs to the Software Engineering degree:**
  - It used to be taught with FIT1008 (in 1st year, 2nd semester)
  - But since 2017, first year Engineering is full with other units
  - Thus, it is now taught in the 2nd year of Software Eng.
  - Main differences:
    - Students have a different background (JavaScript)
    - More emphasis on Software Engineering best practices

Overview Weekly Topics

Preliminary Timetable for Topics

Week	Topics
1	MIPS Architecture and MIPS simple programs.
2	Decisions in MIPS, Arrays in MIPS and Iteration in MIPS.
3	MIPS Memory and MIPS Functions.
4	Complexity, Sorting, Exceptions and Assertions.
5	Abstract Data Types; Classes and Objects and Variables and Scoping.
6	Stacks and Queues based on Arrays.
7	(Sorted and Unsorted) Array-based Lists and on Iterators.
8	Linked Stacks, Queues and Lists.
9	Hashing and Collision Resolution (linear Probing).
10	Recursion, Iteration vs Recursion and Recursive Sorts.
11	Binary Trees and Binary Search Trees.
12	Heaps and Priority Queues.

Coordinated with FIT1045 and FIT2004 for less repetition and more reuse

Coordinated with ENG1003 for more background material

Recommended Reading

Recommended Reading

- **Slides are not enough: read the recommended texts!**
  - Available at https://monash.rl.talis.com/
- **For MIPS:**
  - R.L. Britton - MIPS Assembly Language Programming
- **For data structure and algorithms:**
  - B. Miller and D. Ranum - Problem solving with algorithms and data structures using Python https://interactivepython.org/runestone/static/pythonds/index.html
  - M.T. Goodrich, R. Tamassia, M.H. Goldwasser - Data Structures and Algorithms in Python
- **For Python:**
  - http://docs.python.org/3/tutorial/
  - http://openbookproject.net/thinkcs/python/english3e/ chapters 1-7

Summary

Summary

- **We have seen:**
  - The teaching team
  - What it is(are) the unit(s) all about
  - What will you get from this unit
  - Main differences between FIT1008 and FIT2085
  - Weekly topics
  - Recommended reading