Topic 1

Introduction to Computation

Murdoch
UNIVERSITY

TODAY

Murdoch UNIVERSITY

- What is computation
- Declarative knowledge vs imperative knowledge
- Algorithm
- Expression of an algorithm
- Basic computer architecture
- Programming languages
- Python programming language
- Python program: definitions and commands
- Python interpreter
- How to run Python programs

WHAT DOES A COMPUTER DO?

Murdoch UNIVERSITY

- Fundamentally:
  - performs calculations
    - billions calculations per second!
  - remembers results
    - hundreds of gigabytes of storage!
- What kinds of calculations?
  - built-in to the language
  - ones that you define as the programmer
- Computers only know what you tell them

TYPES OF KNOWLEDGE

Murdoch UNIVERSITY

- declarative knowledge is statements of fact.

  Eg, someone will win a prize today

- imperative knowledge is a recipe or “how-to”.

  1) Participants sign up for prize draw
  2) Organiser opens the app
  3) Organiser chooses a random number between 1st and nth participant
  4) Organiser finds the number in the participant sheet. Winner!

A NUMERIC EXAMPLE

Murdoch UNIVERSITY

- Declarative: square root of a number `x` is `y` such that `y*y=x`
- Imperative: recipe for deducing square root of a number `x` (16)
  1. Start with a **guess**, `g`
  2. If `g*g` is **close enough** to `x`, stop and say `g` is the answer
  3. Otherwise make a **new guess** by averaging `g` and `x/g`
  4. Using the new guess, **repeat** process until close enough

g	g*g	x/g	(g+x/g)/2
3	9	16/3	4.17
4.17	17.36	3.837	4.0035
4.0035	16.0277	3.997	4.000002

WHAT IS A RECIPE?

Murdoch UNIVERSITY

1) sequence of simple steps
2) flow of control specifies when each step is executed
3) a means of determining when to stop

1+2+3 = an algorithm!

ALGORITHM

Murdoch UNIVERSITY

- *Informally*, an algorithm is an ordered sequence of instructions that is *guaranteed* to solve a specific problem. Algorithms are important because if you can specify a working algorithm for a problem then you can:
  - Solve the problem
  - Get a computer to solve any equivalent forms of that problem automatically for you.

- A *formal* definition of an algorithm is:

An Algorithm is a well-ordered collection of unambiguous and effectively computable operations that, when executed, **produces a result** and halts in a finite amount of time.

ALGORITHM contd…

Murdoch UNIVERSITY

*"…**well-ordered**…"*

- The order of the operations that makes up an algorithm must be correct (i.e., steps cannot be in an order which produces the wrong result.)
- There is also often an efficiency aspect to the order of the instructions – that is, one particular order is more efficient than other orders even if they give the same correct result.
- It is also important that the first operation is clearly indicated so that it is clear where the algorithm starts.
- Similarly, its end must be clear, otherwise the algorithm might carry on forever.

ALGORITHM contd…

"…unambiguous…"

So, each instruction must also be clear in what it does

E.g. Do part 1 or part 2

The following is *less* ambiguous but is it completely unambiguous?

if you are an adult do Part 1 else do Part 2

ALGORITHM contd…

Murdoch UNIVERSITY

*"…**effectively computable operations**…"*

- The operation must be within the capabilities of the computer it will be executed on.
- For example, a = b + c is effectively computable only if the computer can perform addition.
- There are also external limits on the sorts of computations that can be performed.
- For example, the expression, a = b / c is only effectively computable when c is not zero.

ALGORITHM contd…

Murdoch UNIVERSITY

*"…**halts in a finite amount of time**…"*

- The algorithm must be capable of finding a result and complete within a finite amount of time.
- Some algorithms may take a very long while to complete for certain sets of data (hours, days, years, centuries...) but they must ultimately be capable of producing a result.
- Repeating Step 4 ("Repeat") on the shampoo bottle would represent what's called an "infinite loop" and so the algorithm would not halt in a finite amount of time.

Step1: wet hair
Step2: apply shampoo
Step3: rinse
Step4: repeat

REPRESENTATION OF AN ALGORITHM

Murdoch UNIVERSITY

- Programmers use different methods of writing down the designs for their algorithms before they translate them into the code for a particular language. Eg
  - Flow charts
  - Structured English
- Here is a rudimentary algorithm for a “menu” program using structured English:

```
display menu
read user input from keyboard
depending on user input do one of the following:
output the 'option 1' message (option 1)
output the 'option 2' message (option 2)
output the 'option 3' message (option 3)
```

REPRESENTATION OF AN ALGORITHM

Murdoch UNIVERSITY

- Pseudocode: an alternative to structured English is *pseudocode* which is written very much like a programming language with minimal syntactic conventions. For example:

```
print "you have 3 choices"
print "enter a for option 1"
print "enter b for option 2"
print "enter c for option 3"

read response

if response == ‘a’
   print "You have selected option 1"
if response == ‘b’
   print "You have selected option 2"
if response == ‘c’
   print "You have selected option 3"
```

STRUCTURE OF AN ALGORITHM

Murdoch UNIVERSITY

- There are only three structures that change the execution flow of an algorithm:
  - Sequence
  - Selection
  - Iteration
- All algorithms are made up of combinations of these three structures and this is known as structure theorem.

COMPUTERS ARE MACHINES

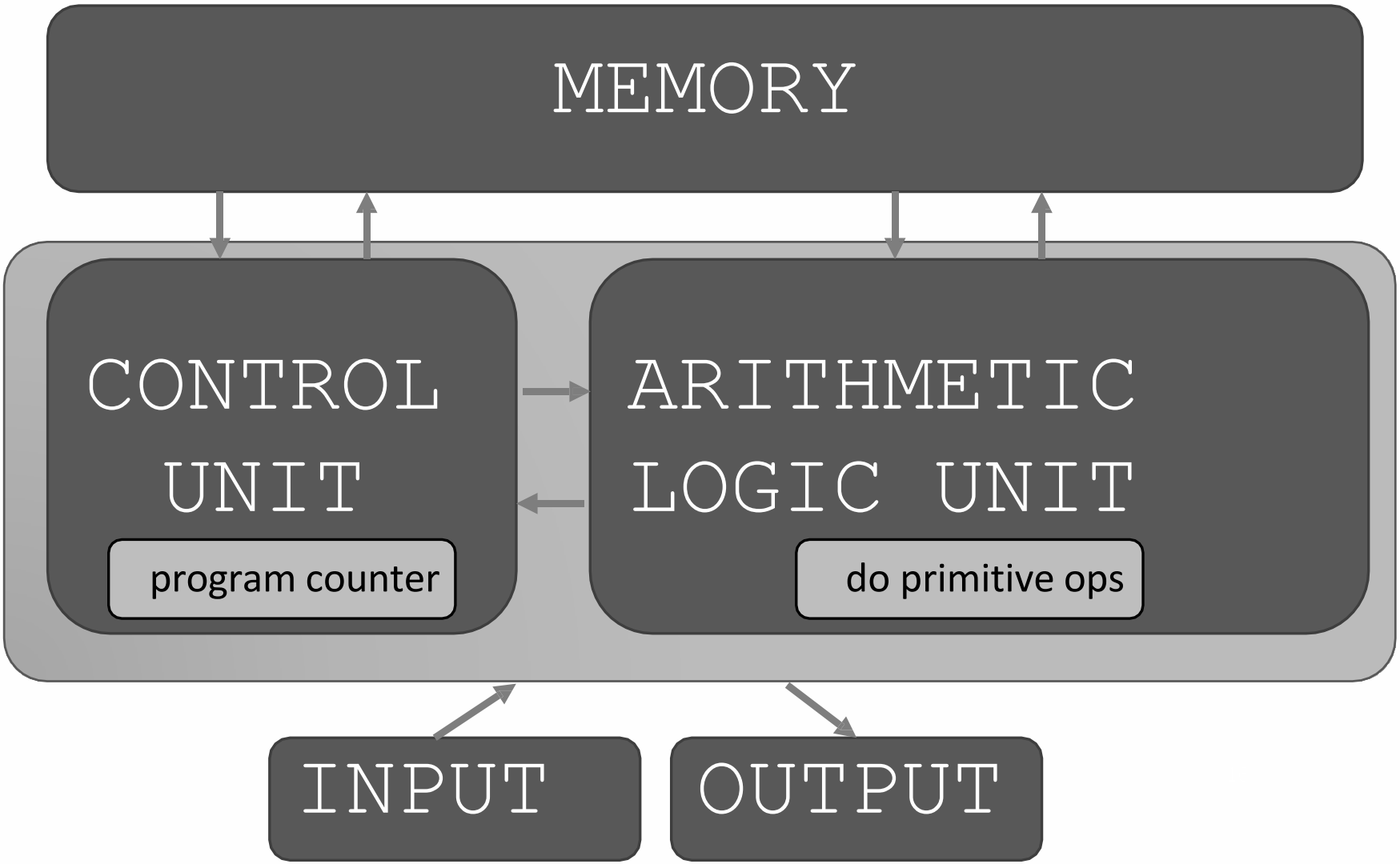
Murdoch UNIVERSITY

- how to capture a recipe/algorithm in a mechanical process
- fixed program computer
  - E.g., a calculator

- stored program computer
  - machine stores and executes instructions
  - programmable

BASIC COMPUTER ARCHITECTURE

Murdoch UNIVERSITY



STORED PROGRAM COMPUTER

Murdoch UNIVERSITY

- sequence of instructions stored inside computer's memory
  - built from predefined set of primitive instructions
    1) arithmetic and logic
    2) simple tests
    3) moving data
- The computer executes each instruction one after the other from the program stored in the computer memory.
- Modern computer systems are controlled by special programs known as operating systems

BASIC PRIMITIVES

Murdoch UNIVERSITY

- Turing showed that you can compute anything using 6 primitives
- modern programming languages have more convenient set of primitives
- can abstract methods to create new primitives
- anything computable in one language is computable in any other programming language

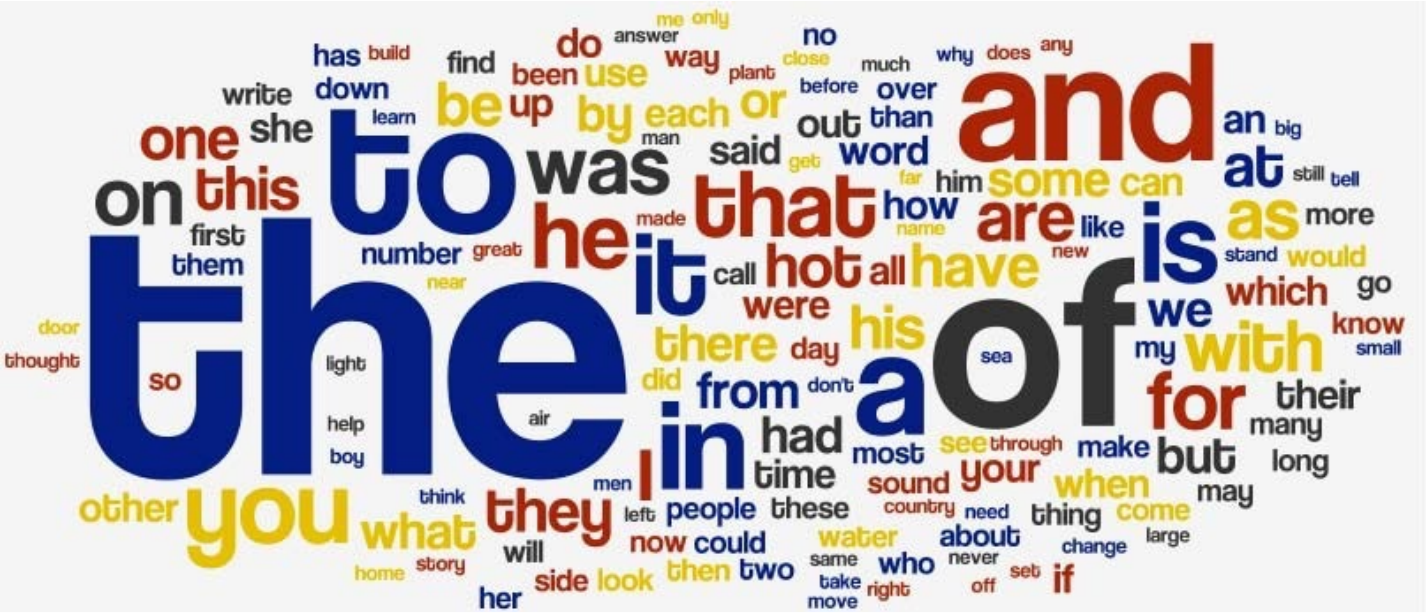
CREATING A RECIPE

Murdoch UNIVERSITY

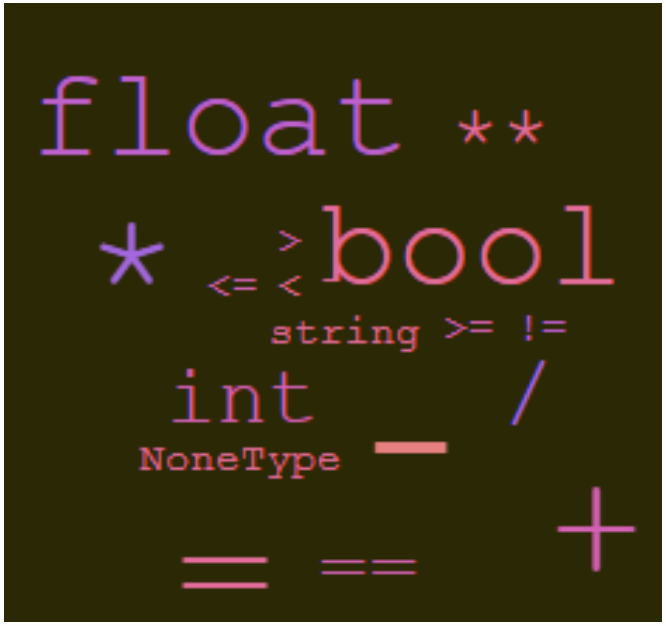
- a programming language provides a set of primitive operations
- expressions are complex but legal combinations of primitives in a programming language
- expressions and computations have values and meanings in a programming language

ASPECTS OF LANGUANGES

- **primitive constructs**
  - English: words
  - programming language: numbers, strings, simple operators



Word Cloud copyright Michael Twardos, All Right Reserved.



Word Cloud copyright unknown, All Right Reserved.

ASPECTS OF LANGUANGES

Murdoch UNIVERSITY

- syntax
  - English:

    `"cat dog boy"` -> not syntactically valid

    `"cat hugs boy"` ->syntactically valid

  - programming language:

    `"hi"5` -> not syntactically valid

    `3.2*5` -> syntactically valid

ASPECTS OF LANGUANGES

Murdoch UNIVERSITY

- static semantics is about which syntactically valid strings have meaning
- English:
  - `"I are hungry"`-> syntactically valid
  - but static semantic error
- Programming language:
  - `3.2*5` -> both syntactically and static semantically valid, as it follows the syntax: *`operand operator operand`*, and both operands for operator * are numbers
  - `3+"hi"`
    - syntactically valid, as it follows the same syntax of *`operand operator operand`*
    - but static semantically wrong, because with operator `+`, both operands `3` and `"hi"` are not of the same type.

ASPECTS OF LANGUANGES

Murdoch UNIVERSITY

- semantics is the meaning associated with a syntactically correct string of symbols with no static semantic errors
  - English: can have many meanings `"Flying planes can be dangerous"`
    1. Flying planes (as opposed to taking other types of transport such as train) can be dangerous
    2. Driving planes (as opposed to driving other types of vehicles such as cars) can be dangerous
  - programming languages: have only one meaning but may not be what programmer intended

WHERE THINGS GO WRONG?

Murdoch UNIVERSITY

- syntactic errors
  - common and easily caught
- static semantic errors
  - syntactically correct but violates some rules or constraints
  - some languages check for these before running program
  - can cause unpredictable behavior
- no semantic errors but different meaning than what programmer intended
  - program crashes, stops running
  - program runs forever
  - program gives an answer but different than expected

PYTHON PRAGRAMMING LANGUAGE

Murdoch UNIVERSITY

- Python is a programming language
  - It is easy to learn
  - It is imperative - you use Python language to tell the computer how to do things
  - It is popular and is widely used, particularly in data science and artificial intelligence
  - There are several versions of the language. The current version is version 3 – Python3

COMPILED LANGUAGE VS INTERPRETED LANGUAGE

Murdoch UNIVERSITY

- Two types of programming languages: compiled and interpreted.
- Compiled language:
  - the program written in the programming language (source code) must be translated into a program expressed as a sequence of machine instructions of the underlying computer.
  - The resulting program is called "executable", which is run directly on the computer system.
  - Generally speaking, this type of programs are more efficient, but it requires an extra step – compilation of the source code program
  - Examples: C, C++, Java, C#, etc

PYTHON IS INTERPRETED

Murdoch UNIVERSITY

- Python is an interpreted programming languages. A special program (known as "interpreter") executes each statement or command in the Python source code.
  - no extra compilation step is required. The program written in Python can be run immediately
  - However, the Python program is not executed directly on the underlying computer system. Instead, it is executed by the Python interpreter. This kind of program execution is known as interpretation.
  - Generally speaking, interpreted programs are not as efficient as the compiled programs.
  - Recent advancement in hardware and interpreter design has minimised the performance difference.

PYTHON PROGRAM

Murdoch UNIVERSITY

- a program is a sequence of definitions and commands
  - definitions **evaluated**
  - commands **executed** by Python interpreter in a shell
- commands (statements) instruct interpreter to do something
- can be typed directly in a shell or stored in a file that is read into the interpreter and executed

PYTHON INTERPRETER

Murdoch UNIVERSITY

- To run a Python program, you need Python program interpreter
  - This is a program that reads in your Python program and execute it.
- There are two ways you can use Python interpreter to run your program:
  - Store your program in a file and run the Python interpreter which will read in your program from the file and execute your program line by line.
  - Directly type your Python program from the Python shell interactively.

PYTHON IDE

Murdoch UNIVERSITY

- Alternatively, you may use an IDE (Integrated Development Environment) to develop your Python program
- An IDE, such as IDLE, allow you to edit, run and debug your program within the same program, simplifying the program development.
- In this unit, we will be using the Python interpreter directly within a terminal window (eg, Command Prompt, PowerShell or Terminal on Windows, terminal on macOS and Linux) to develop our programs.
- We also use an IDE, namely IDLE, to develop our program.

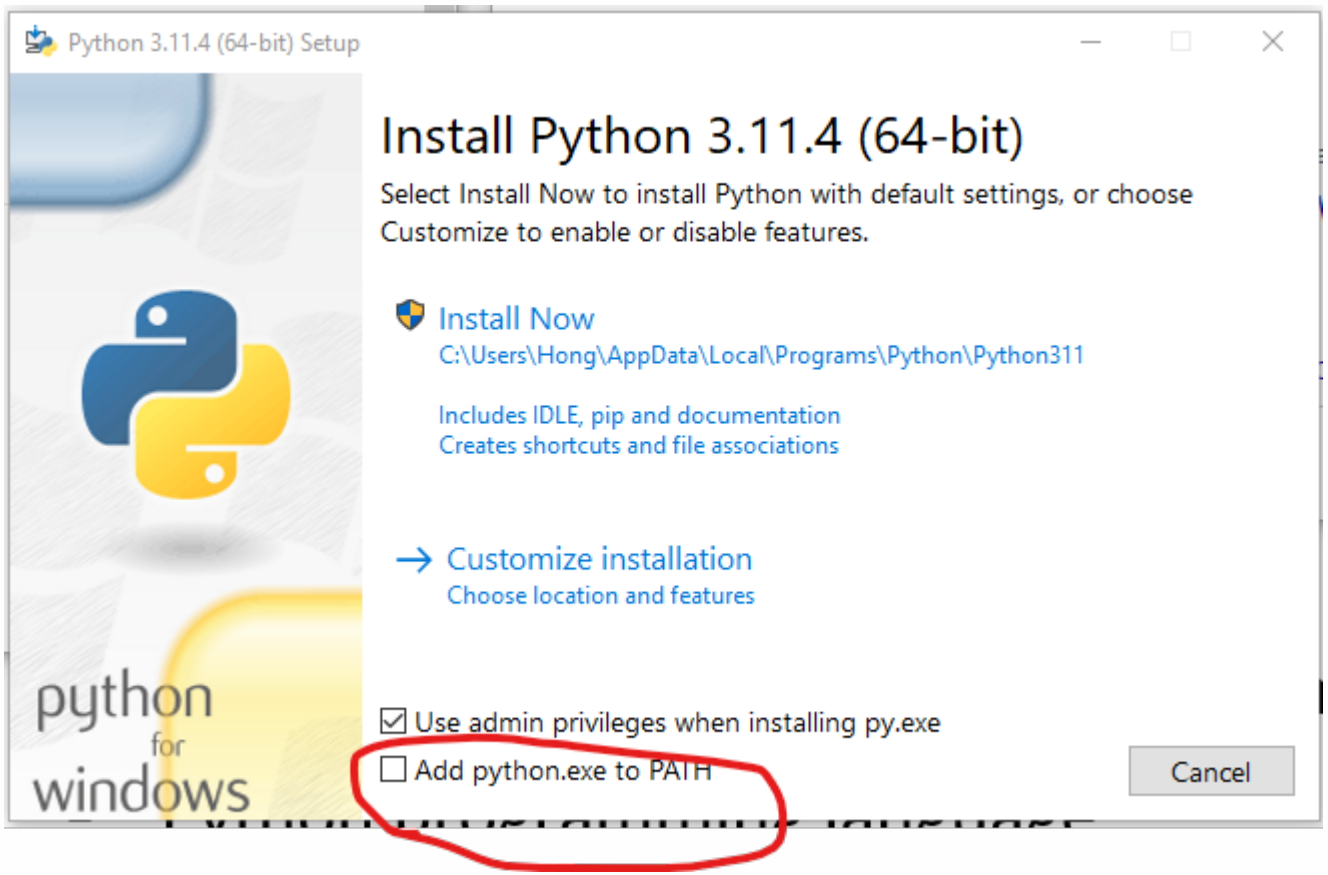
PYTHON3 PACKAGE

Murdoch UNIVERSITY

- You can download and install the latest version of Python interpreter from https://www.python.org/downloads/
- The download includes the latest version of Python interpreter (version 3.12). It also includes a number of development tools :
  - Idle (Integrated Development and Learning Environment) – a simple IDE bundled with the Python interpreter in the same download
  - pip – package manager for Python packages and
  - pydoc – which generates Python documentation from Python modules.

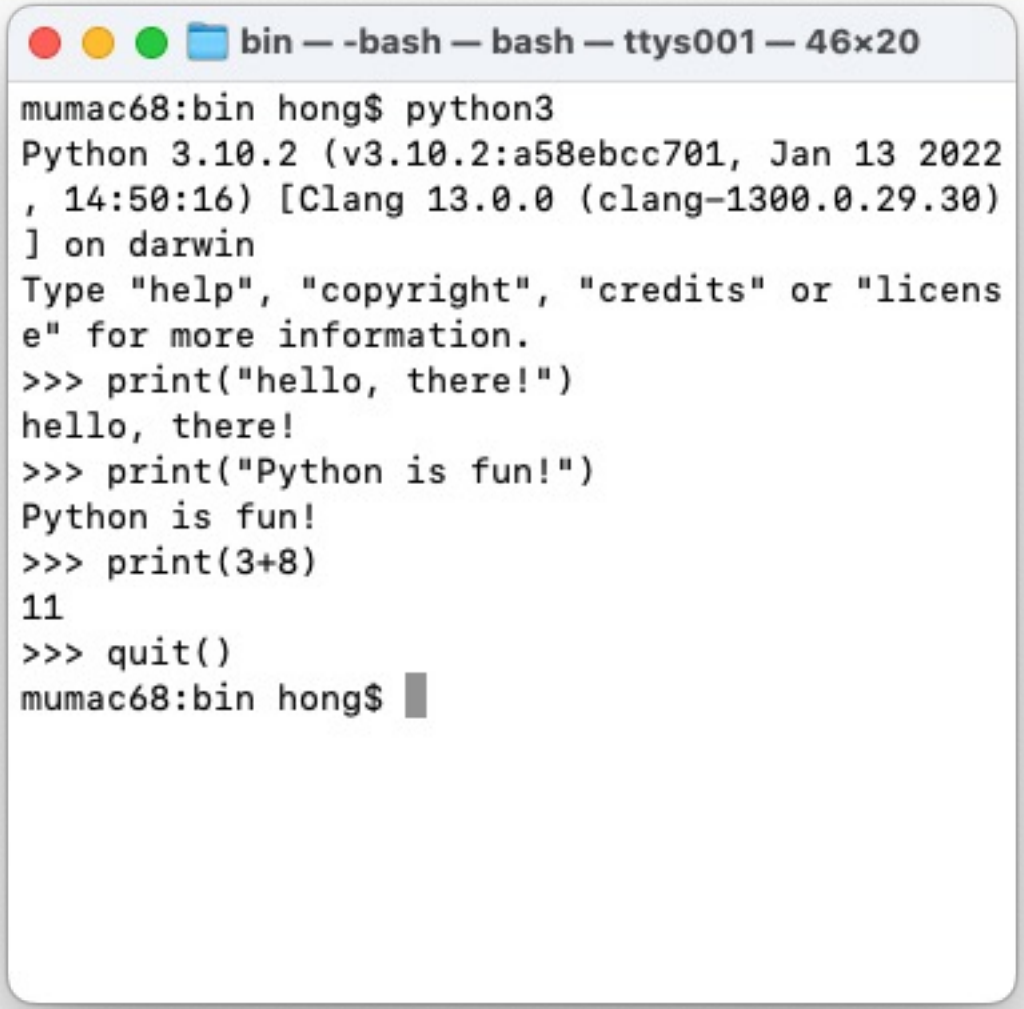
INSTALL PYTHON

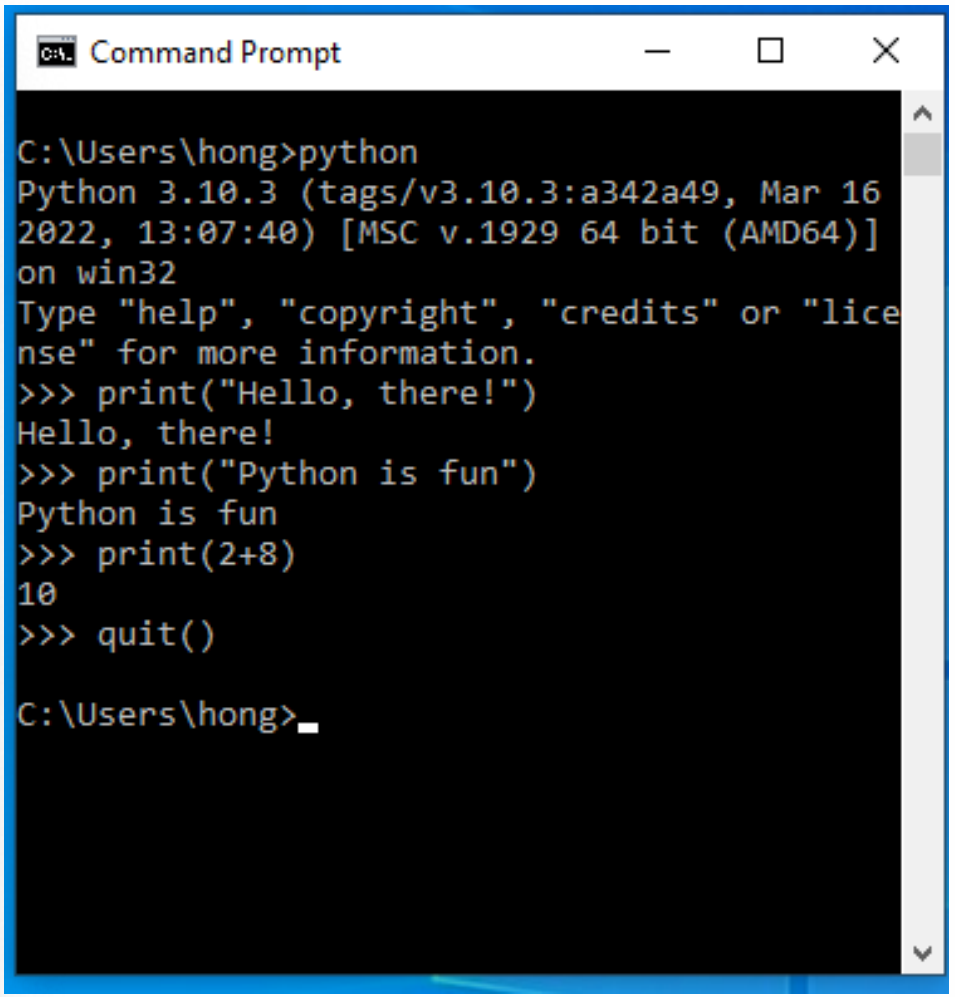
Murdoch UNIVERSITY



RUN PROGRAM FROM PYTHON SHELL

Murdoch UNIVERSITY





Type and run program from Python shell:
Terminal on macOS (left) and Command Prompt (right)

RUN PYTHON PROGRAM STORED IN A FILE

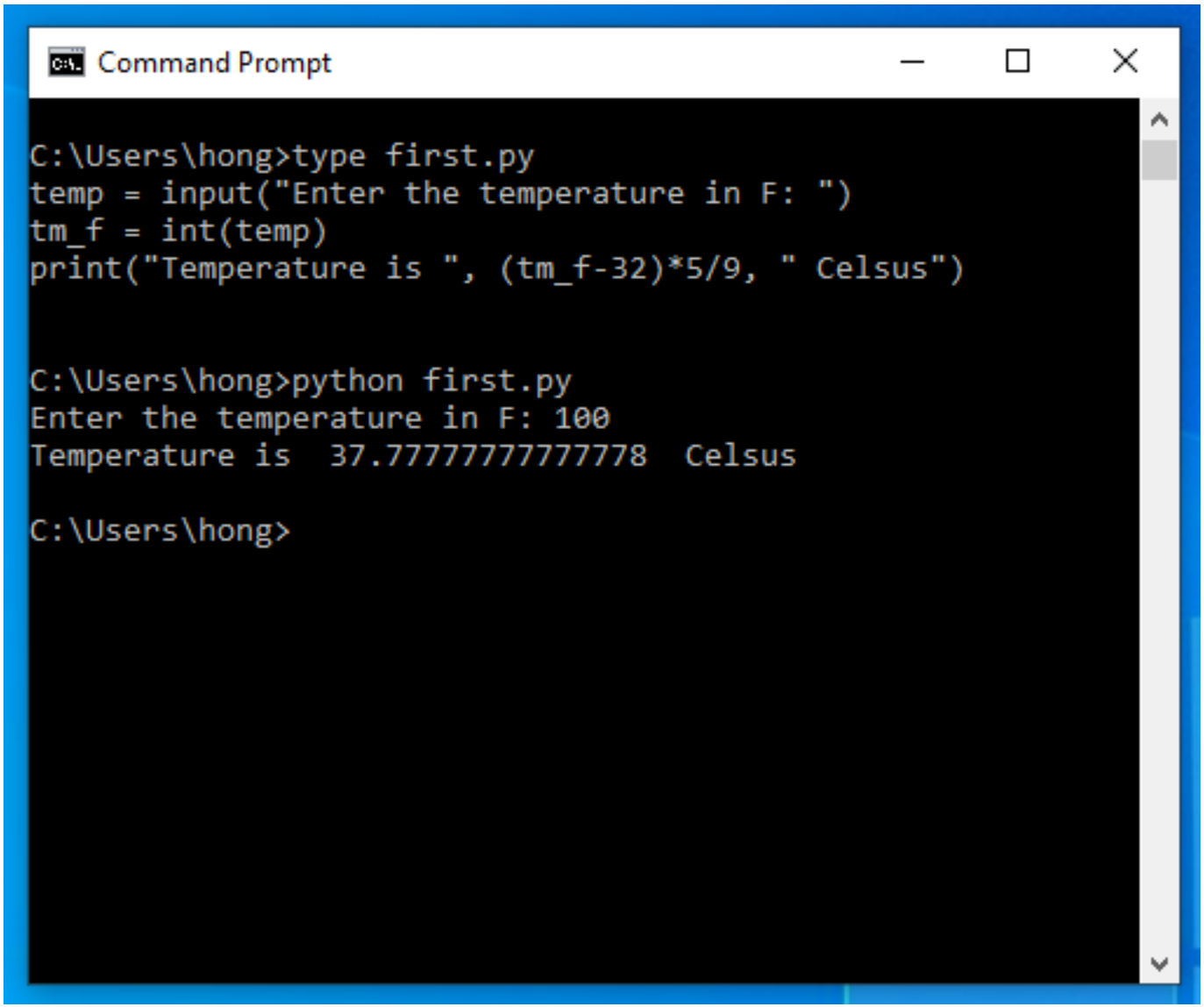
Murdoch UNIVERSITY

Python program stored in file `first.py`:

```
name = input("Type your name: ")
print("my name is " + name)
```

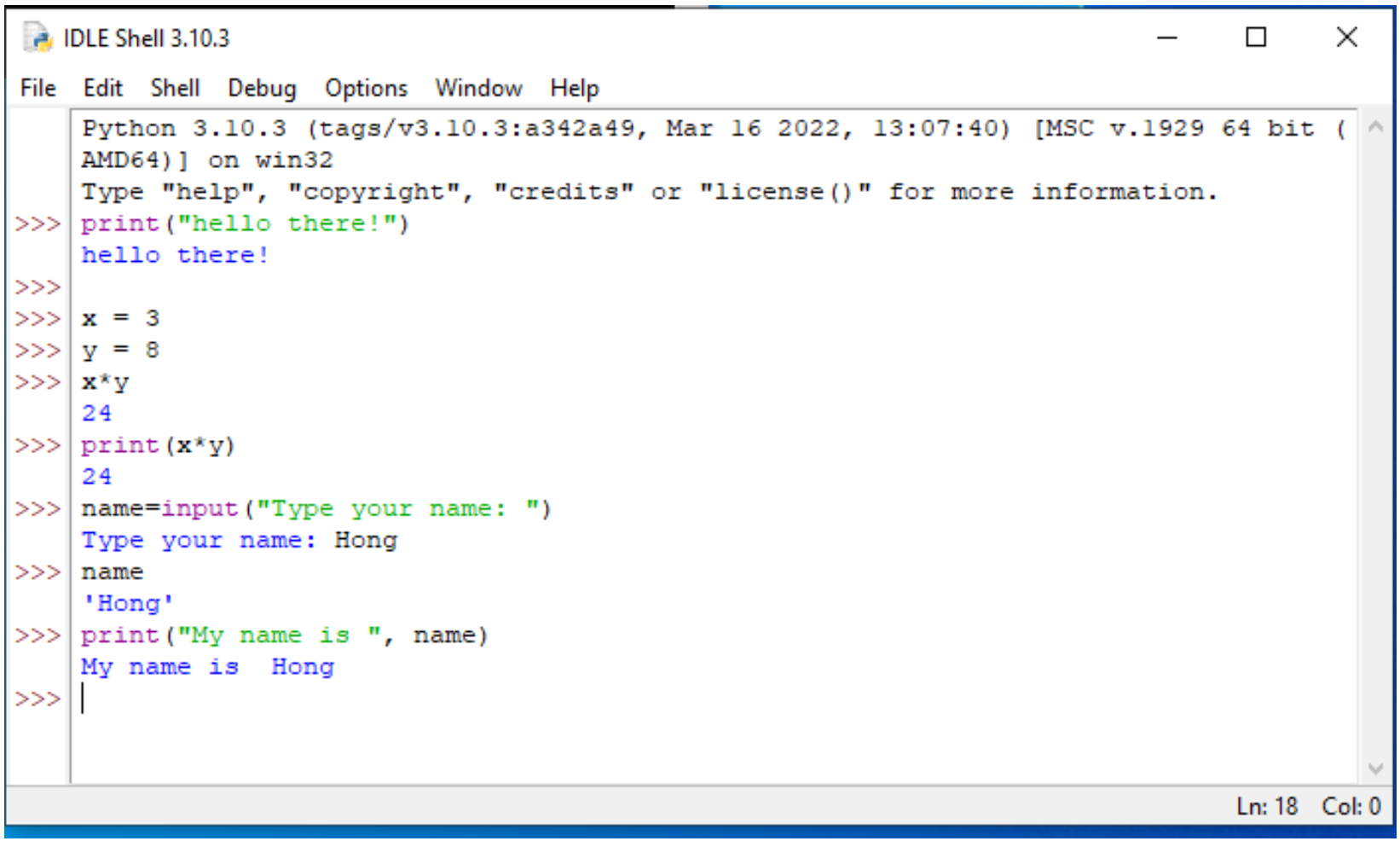
RUN PYTHON PROGRAM STORED IN A FILE

Murdoch UNIVERSITY

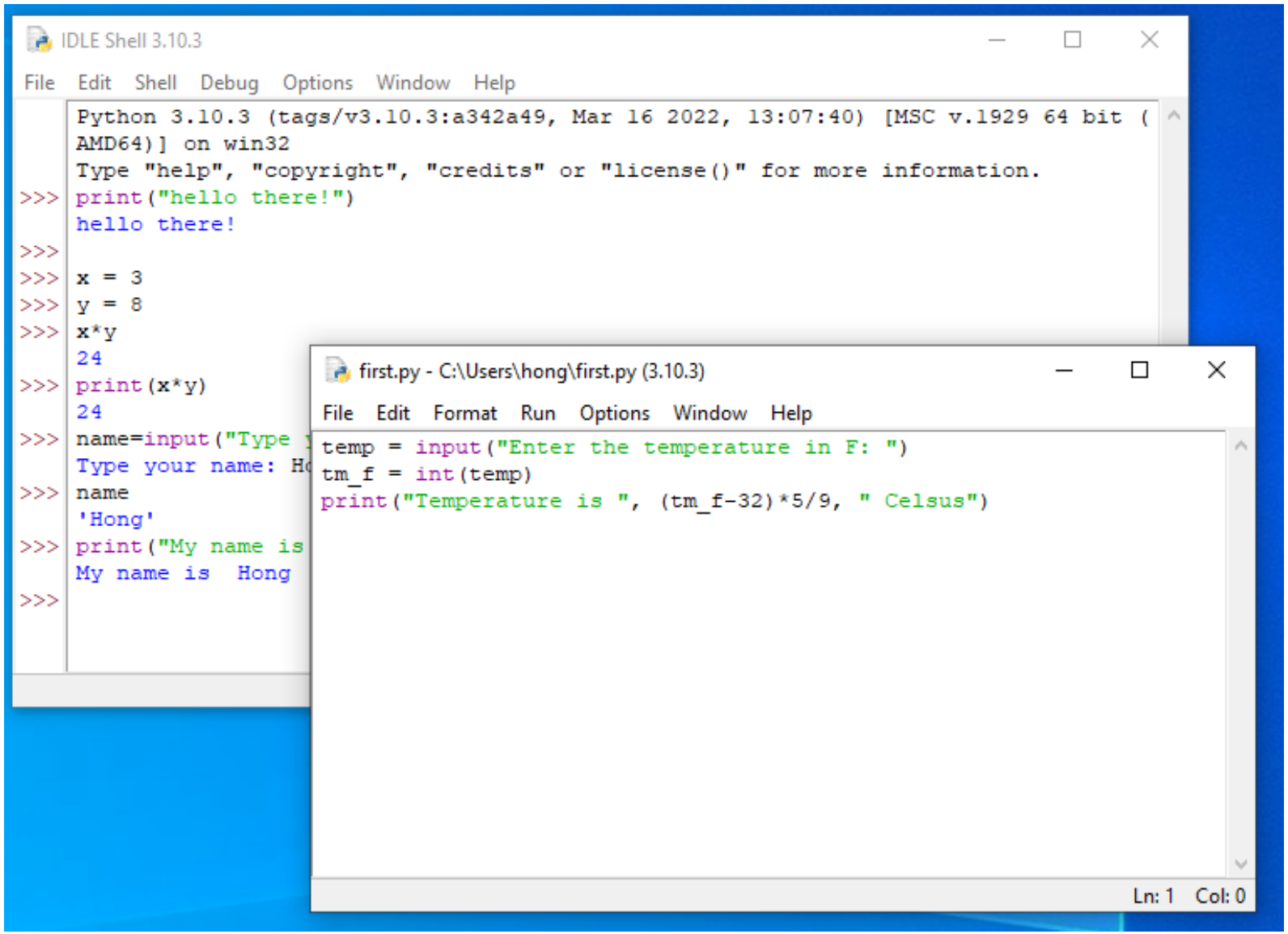


DEVELOP PROGRAM IN IDLE

Murdoch UNIVERSITY



DEVELOP PROGRAM IN IDLE



Use *File* menu to open Python file `first.py`. Run the program using *Run menu*.

SUMMARY

Murdoch UNIVERSITY

- What is computation
- Algorithms and how to express them
- Stored program computer and basic computer architecture
- Programming language syntax and semantics
- Compiled language vs interpreted language
- Python programming language
- How to run a Python program

ACKNOWLEDGEMENT

Murdoch UNIVERSITY

- Sources used in this presentation include:
- `Programiz.com`
- `MIT OCW`