PGR107
Python Programming

Lecture 1 – Introduction

Welcome!

- Welcome to PGR107.
- Instructor is Hadi Zahmatkesh.
- This is a Programming course with Python.
- Intended for people with no experience with programming.
- Please ask questions/let me know if I am difficult to understand.
- 1 x 2 hours lecture + 1 x 2 hours exercise session.

Course Requirements & Assessment

- Submission of one programming project

- Final written exam

Course Materials & Project!

- All the slides that I will talk about during lectures will be up on Canvas before lectures.

- Project will be posted on Canvas too.

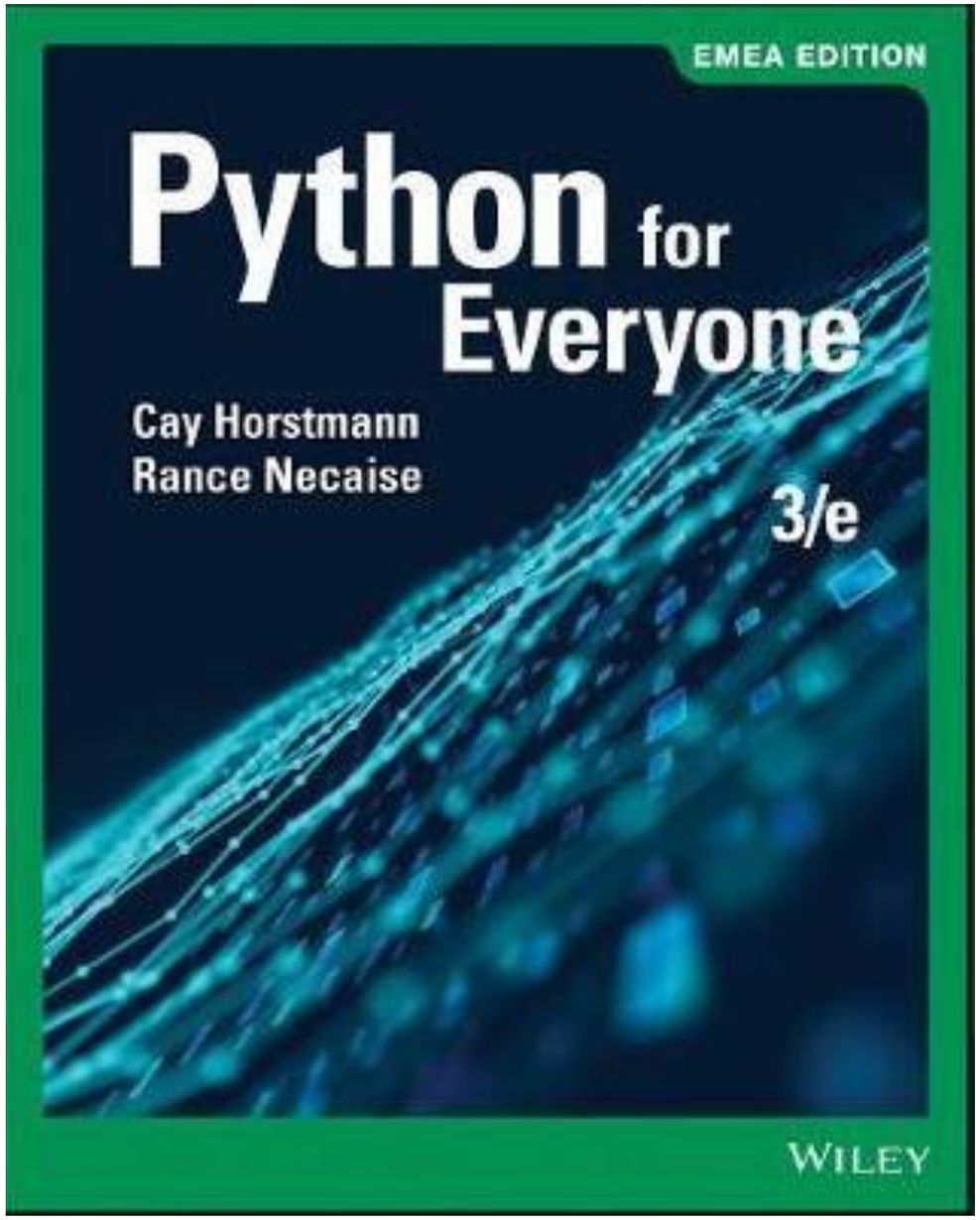
- Project will be done in a group of (1-5) students.

- In exercise sessions, you will write Python programs related to the topics we discussed in the lectures.

The Book!

- Python for Everyone by Cay Horstmann and Rance Necaise (2nd or 3rd Edition)

- Can get it easily on Akademika bookstore.



Getting Help

- You can always email me. Please have **PGR107** in the title.

hadi.zahmatkesh@oslomet.no

Outline

- Chapter 1 – Introduction
- Chapter 2 – Programming with Numbers and Strings
- Chapter 3 – Decisions
- Chapter 4 – Loops
- Chapter 5 – Functions
- Chapter 6 – Lists and Tuples
- Chapter 8 – Sets and Dictionaries
- Chapter 7 – Files and Exceptions
- Chapter 9 – Objects and Classes
- Chapter 10 – Inheritance

Chapter 1 - Introduction

Chapter Goals

- To learn about computers and programming
- To write and run your first python program
- To recognize compile-time and run-time errors
- To describe an algorithm with pseudocode

Computer Programs

- **Computer:** A machine that stores data, interacts with devices, and executes programs.

- **Computer program:** Tells a computer a sequence of steps that are needed to fulfill a task.
  - ✓ The physical computer and peripheral devices are collectively called **hardware**.
  - ✓ The programs the computer executes are called **software**.

- **Programming:** The act of designing and implementing computer programs (giving instructions to computers).

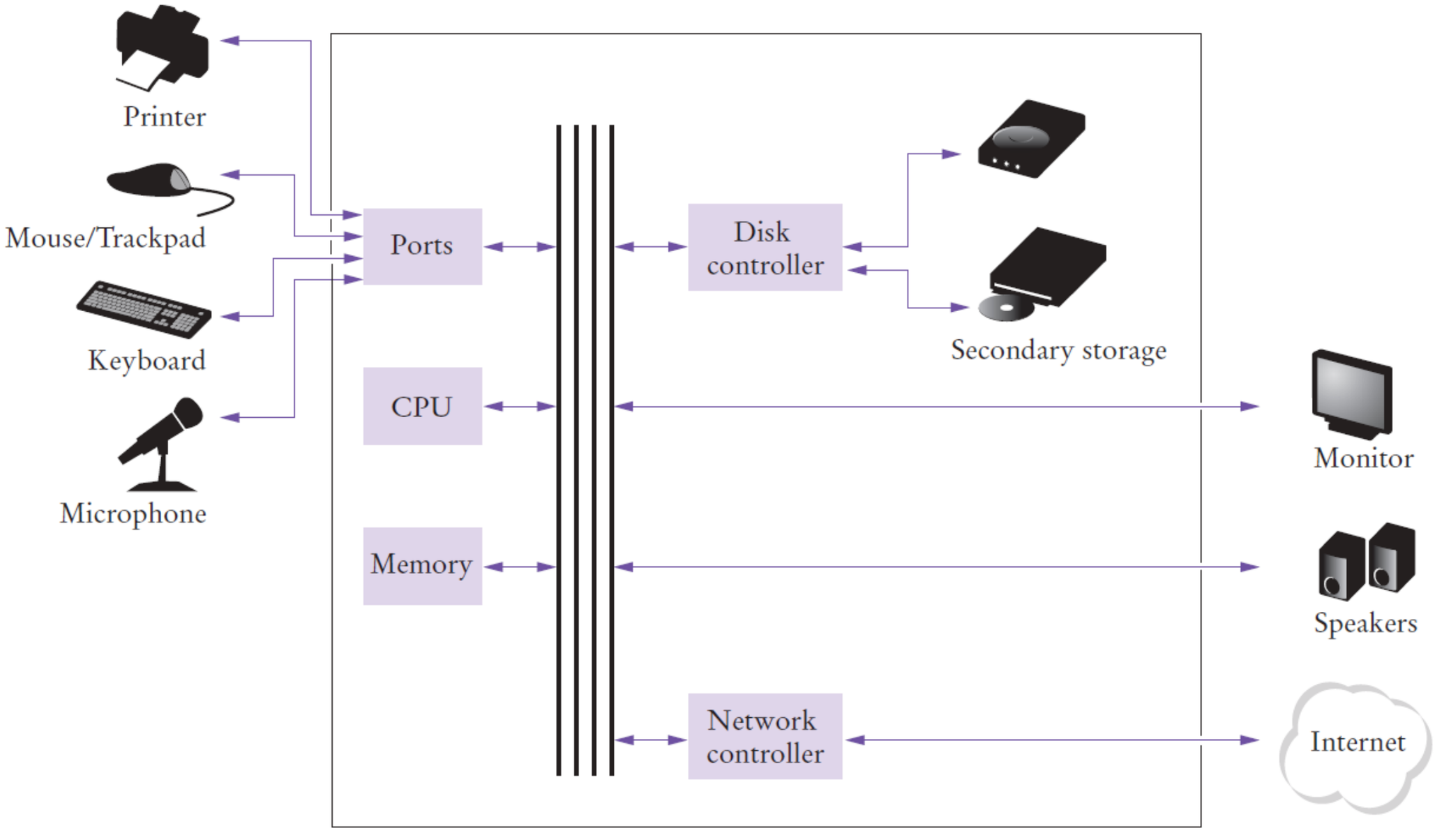
The Anatomy of a Computer

- **Central Processing Unit (CPU):** Performs program control and data processing. That is, the CPU locates and executes the program instructions.

- **Storage devices:** Include primary storage and secondary storage
  - ✓ **Primary storage:** is made from memory chips (electronic circuits that can store data provided that they are supplied with electric power).
  - ✓ **Secondary storage** (usually a **hard disk**): Provides slower and less expensive storage that persists without electricity.

- **Peripheral devices:** Used to interact with a human user
  - ✓ **Input devices:** keyboard, mouse, Microphone
  - ✓ **Output devices:** printer, speaker, display screen

Schematic Design of a Personal Computer



The Python Programming Language

- Developed by Guido Van Rossum in the early 1990s.
- High-level programming language.
- Much simpler and cleaner syntax than other popular languages such as C, C++, Java.
- Portable and easier to learn and use.

How to Get Started With Python

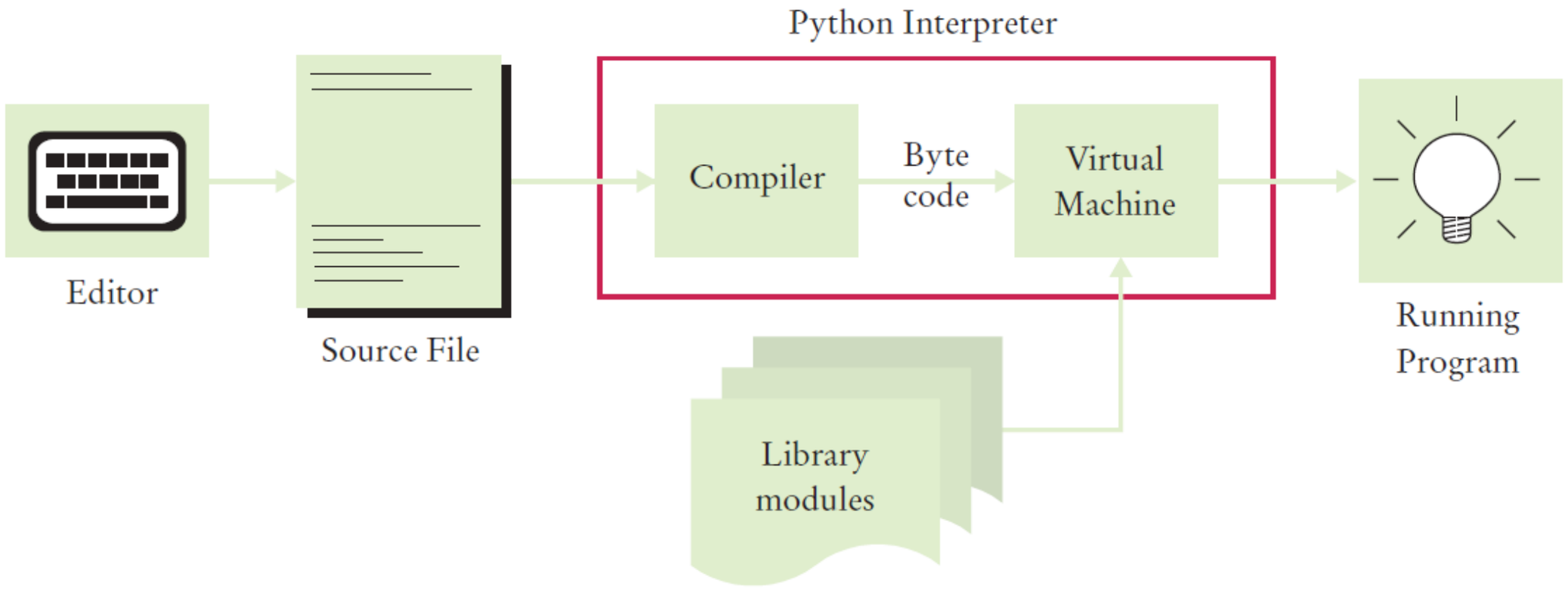
- Install Python and an IDE (Integrated Development Environment)

- **Anaconda:** A free and open-source distribution of Python programming language.

- **Spyder:** An open-source IDE for scientific programming in the Python language.

- Now you are ready to start!

From Source Code to Running Program



Analyzing Your First Program

- A Python program contains one or more lines of instructions or **statements** that will be translated and executed by the Python interpreter.

```
1  # My first Python program.
2  print("Hello, World!")
```

- The first line is a **comment**. Comments begin with # and are not statements.

- The second line contains a statement. That prints or displays a line of text, namely "**Hello, World!**".

Print Statement

Syntax

```
print()
print(value_1, value_2, ..., value_n)
```

All arguments are optional. If no arguments are given, a blank line is printed.

```
print("The answer is", 6 + 7, "!")
```

The values to be printed, one after the other, separated by a blank space.

Errors

- Programmers spend fair amount of time fixing errors in their codes.

- **Compile-time error**: is a violation of the programming language rules that is detected when a code is translated into executable form. Compile-time errors are sometimes called **syntax errors**.

```
File "hello.py", line 2
   print("Hello, World)
                       ^
SyntaxError: EOL while scanning string literal
```

Errors

- **Run-time error**: is any error that occurs when the program compiles and runs, but produces unexpected results.

- Suppose the following:
- **print (1 / 0)**

```
Traceback (most recent call last):
   File "hello.py", line 3, in <module>
ZeroDivisionError: int division or modulo by zero
```

Errors

- **Unfortunately,** sometimes the interpreter is not very smart and often provides no help in identifying the errors (specially the **syntax errors**).

- For example:

- **print (Hello, World!)**

- The error report looks like this:

```
File "hello.py", line 2
    print(Hello, World!)
                       ^
SyntaxError: invalid syntax
```

Problem Solving: Algorithm Design

- Before a program is written, a programmer must clearly understand **what** the date (**input**) is to be used, the desired result or **output**, and the **steps** to be used to produce the result.

- The steps selected to produce the result is referred to as an **algorithm**.

- When English phrases are used to describe the algorithm (**the processing steps**), the description is called **pseudocode**.

Algorithm - Pseudocode

- **Example:** Sum of first 100 integers
  - ✓ Set n equal to 100
  - ✓ Calculate sum using the formula **sum = n (n + 1) / 2**
  - ✓ Display the sum

- **Case Study:** Design and Develop
  - The circumference, C, of a circle is given by the formula C = 2 * PI * r, where PI is the constant 3.14, and r is the radius of the circle. Using this information, write a program to calculate the circumference of a circle that has 2-inch radius.

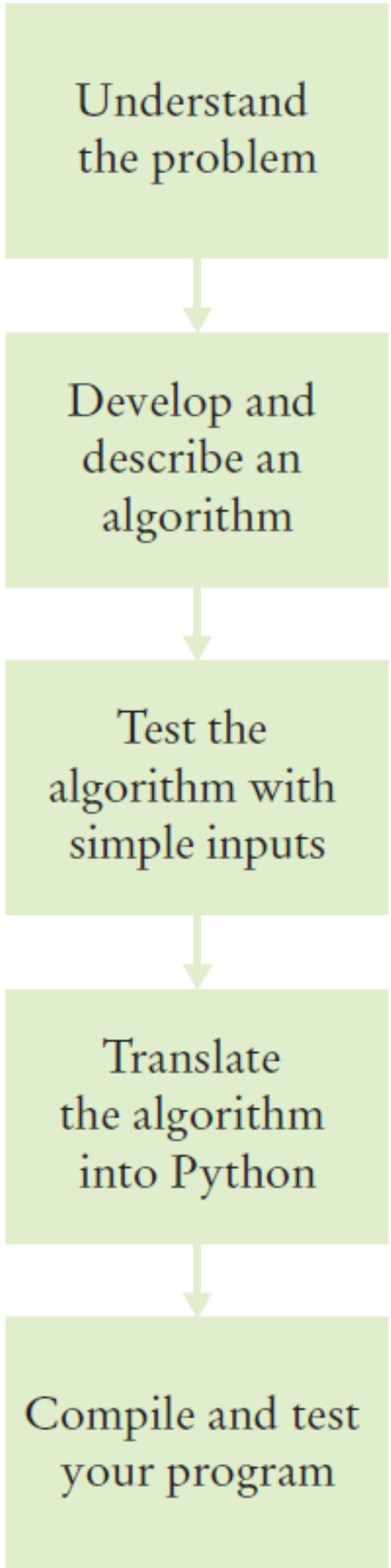
Case Study: Design and Develop

- Analyze the problem:
  - output: circumference (c)
  - Input: radius (r)
  - Formula: C = 2 x PI x r

- Select an algorithm:
  - Set the radius value (r) to 2
  - Calculate the circumference (C), using the given formula
  - Display the calculated C value.

- Write the program now.

You need to first discover and describe an algorithm for the task that you want to solve before you start programming.



Python 2 vs. Python 3

- A programming language is constantly evolving.
- The creator of the language decided to create a new version of Python in 2008.
- If you update to Python 3, your Python 2 code is not going to work.

- print ‘Welcome to MEK1300’ #Python 3.x doesn’t support
- print (‘Welcome to MEK1300’)

- Python 3 is the way to go, is the way of future.

End of Chapter 1