HO CHI MINH CITY UNIVERSITY OF TRANSPORT
FACULTY OF INFORMATION TECHNOLOGY
SOFTWARE ENGINEERING DEPARTMENT

CHAPTER 4
INTRODUCTION TO PROGRAMMING AND PYTHON

CONTENTS

1. Programs and Programming

2. Programming Language

3. Introduction to Python

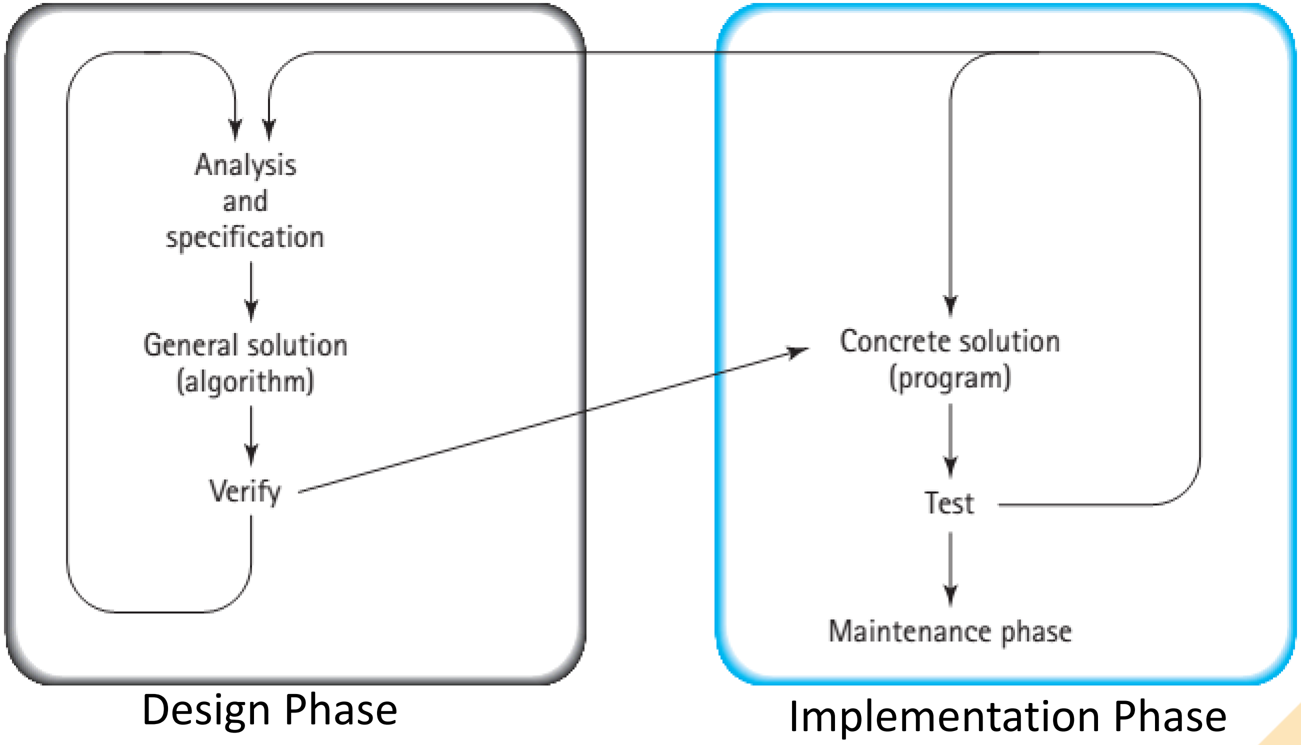
4. Guideline for Installing and Using Anaconda on Windows

1. Programs and Programming

- This course is about programming and computational thinking, not about learning a programming language
- Once you learnt programming in one language, it is relatively easy to learn another language, such as C++, Java,…
- **What is programming?**
  - Programming is writing computer code to create a program, to solve a problem.
  - Programming is the translation of algorithm into a program to tell a computer exactly what to do and how to do.
- **Why learn programming?**
  - Programming gives you the ability to digitize your ideas.
  - It is the important skill for your job

What is a program?

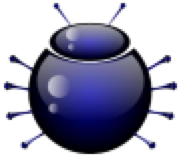
- A **program** is a sequence of instructions that performs a specific task when executed by a computer.
- A program implements an algorithm.
- Program created by a programmer using a programming language.



What is a program?

- A few basic instructions in the program:
  - **input**: Get data from the keyboard, a file, the network, or some other device.
  - **output**: Display data on the screen, save it in a file, send it over the network, etc.
  - **math**: Perform basic mathematical operations.
  - **conditional execution**: Check for certain conditions and run the appropriate code.
  - **repetition**: Perform some action repeatedly.

What is debugging?

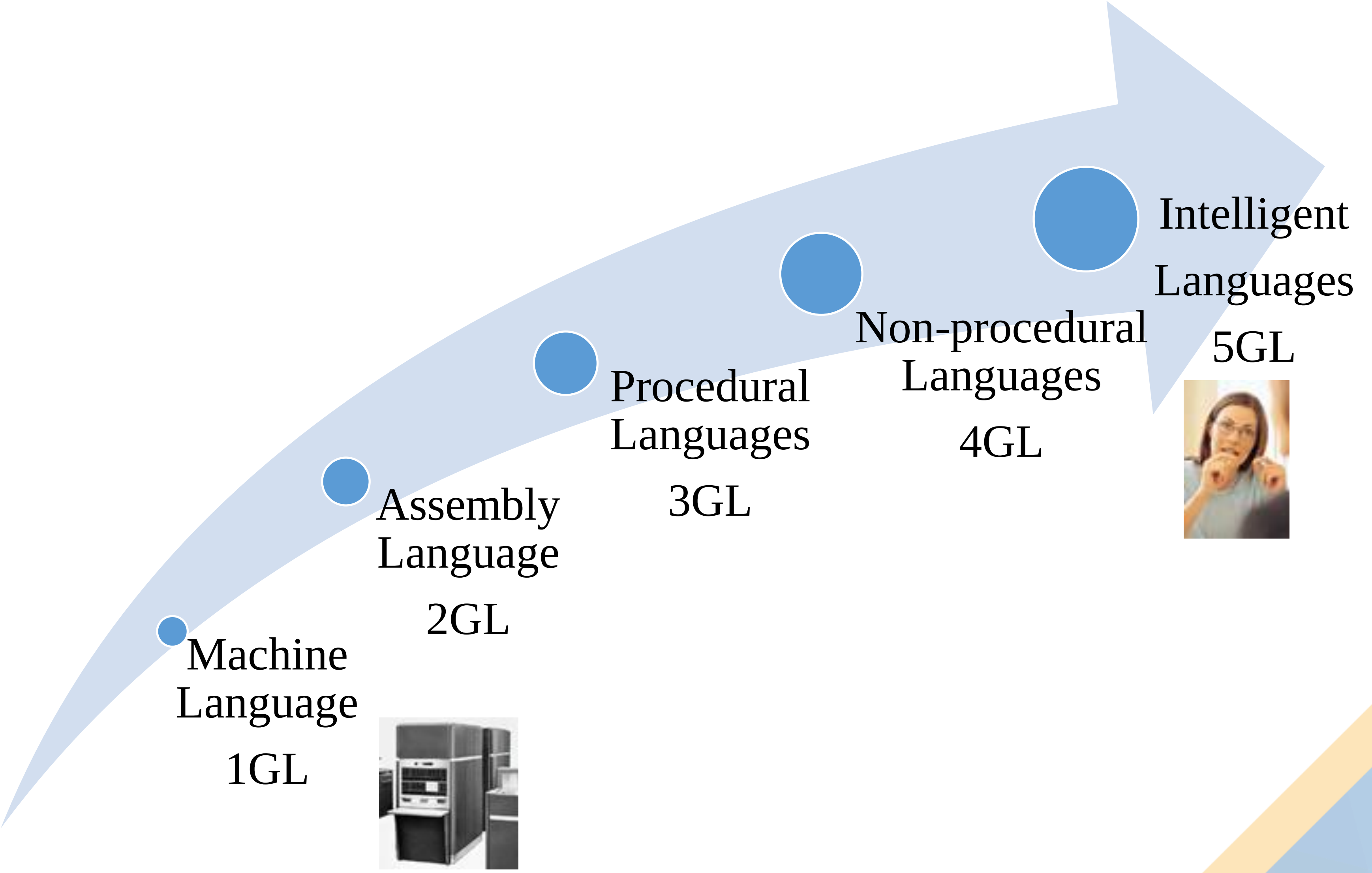


- A **bug** is a mistake in a program.
  - Bug may cause a program crash or give incorrect information.
- **Debugging** means to find the mistake and to fix it.
- Kinds of errors:
  - **Syntax error**: Python cannot understand your program, and refuses to execute it.
  - **Runtime error**: When executing your program (at runtime), your program suddenly terminates with an error message.
  - **Semantic error**: Your program runs without error messages, but does not do what it is supposed to do.

2. Programming Language

- An language used to create programs.
- Have a set of keywords, syntax, rules for instructing a computer to perform specific tasks.
- The most popular languages are: Python, Java, Javascript, C/C++,…
- **Levels of Language**
  - Depending on how close they are to the language the computer itself uses (0s and 1s = low) or to the language people use.

Generation of programming languages



Machine Language

1GL

Assembly Language

2GL

Procedural Languages

3GL

Non-procedural Languages

4GL

Intelligent

Languages

5GL

Levels: Overview

Low Level	• Machine specific • Fast execution • Difficult programming and debugging • Long code	Machine Language (1GL)
		Assembly Language (2GL)
High Level	• Human language like • Slower execution • Easier programming and debugging • Shorter code	Procedural Languages (3GL)
		Non-Procedural Languages (4GL)
		Intelligent Languages (5GL)

1GL: Machine Language

- The only language that the computer directly understands.
- Represents data and program instructions as 1s and Os-binary digits.
- Each type of computer has its own machine language.

  →A computer could not understand programs written in another machine language.
- **Advantages**:
  - The code can run very fast and efficiently, since it is directly executed by the CPU
- **Disadvantages:**
  - Difficult to read, write, and edit if errors occur.
  - Hardware-dependent programming language
  - Not portable.

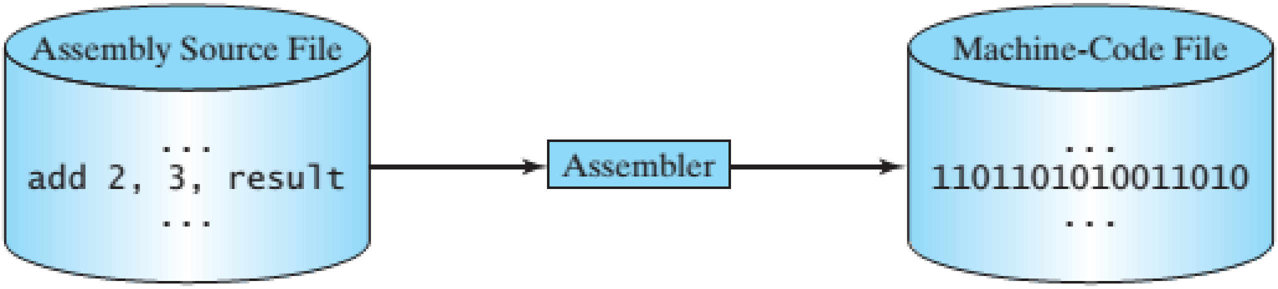
1GL: Machine Language

- **Example**: Machine Language To Add Two Numbers

Location	**Instruction Code**	**Instruction Code**	**Instruction**	**Comments**
Hex	**Binary**	**Hex**		
100	0010 0001 0000 0100	2104	LDA 104	Load first operand into AC
101	0001 0001 0000 0101	1105	ADD 105	Add second operand to AC
102	0011 0001 0000 0110	3106	STA 106	Store sum in location 106
103	0111 0000 0000 0001	7001	HLT	Halt computer
104	0000 0000 0101 0011	0053	operand	83 decimal
105	1111 1111 1111 1110	FFFE	operand	-2 decimal
106	0000 0000 0000 0000	0000	operand	Store sum here

2GL: Assembly Language

- Is a symbolic language that use symbols to represent machine-language instructions.
- An assembly language statement consists of a label, an operation code, and one or more operands.
- Closely connected to machine language and the internal architecture of the computer system.
- A **source program** written in assembly language need a translator often known as the **assembler** to convert them into machine language.



2GL: Assembly Language

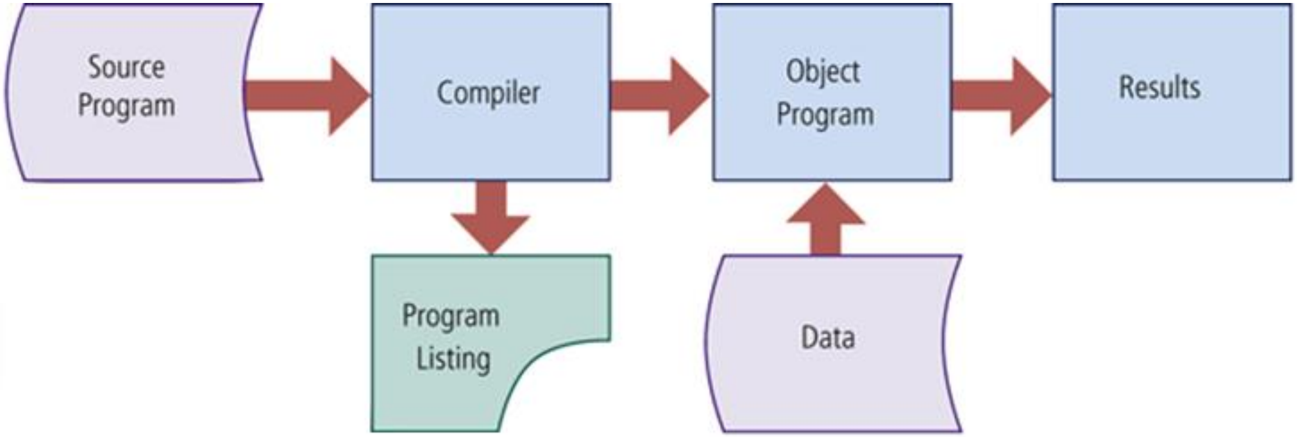
- **Example**: Assembly Language program to add two numbers.

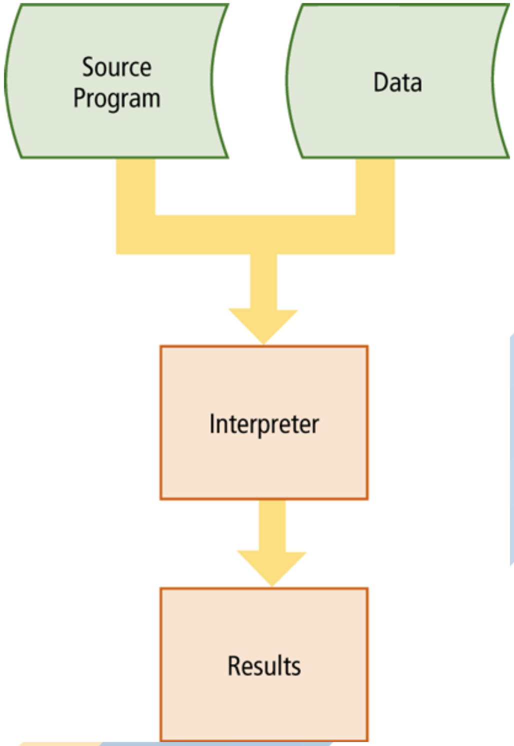
```
            ORG 100     /Origin of program is location 100
            LDA A       /Load operand from location A
            ADD B       /Add operation form location B
            STA C       /Store sum in location C
            HLT         /Halt computer
A,          DEC 83      /Decimal operand
B,          DEC –2      /Decimal operand
C,          DEC 0       /Sum stored in location C
            END
```

- **Advantages:**
  - Efficient in terms of execution time and main memory usage
- **Disadvantages:**
  - Machine-dependent → less portable

3GL: Procedural Languages

- The program statements are not closely related to the internal characteristics of the computer and is referred to **high-level languages.**
- Resembles natural/human languages.
- A **translator** is needed to translate the instructions written in high level language into machine language.
  - Interpreter
  - Compiler





3GL: Procedural Languages

- Some procedural languages**:** Pascal, C/C++, C#, Java, Cobol,…
- **Advantages:**
  - Portable
  - Easy to read, write, debug → the programmer has more time to think about overall program logic.
- **Disadvantages:**
  - Require translator → slow execution.
- **Example**: C program to add two numbers

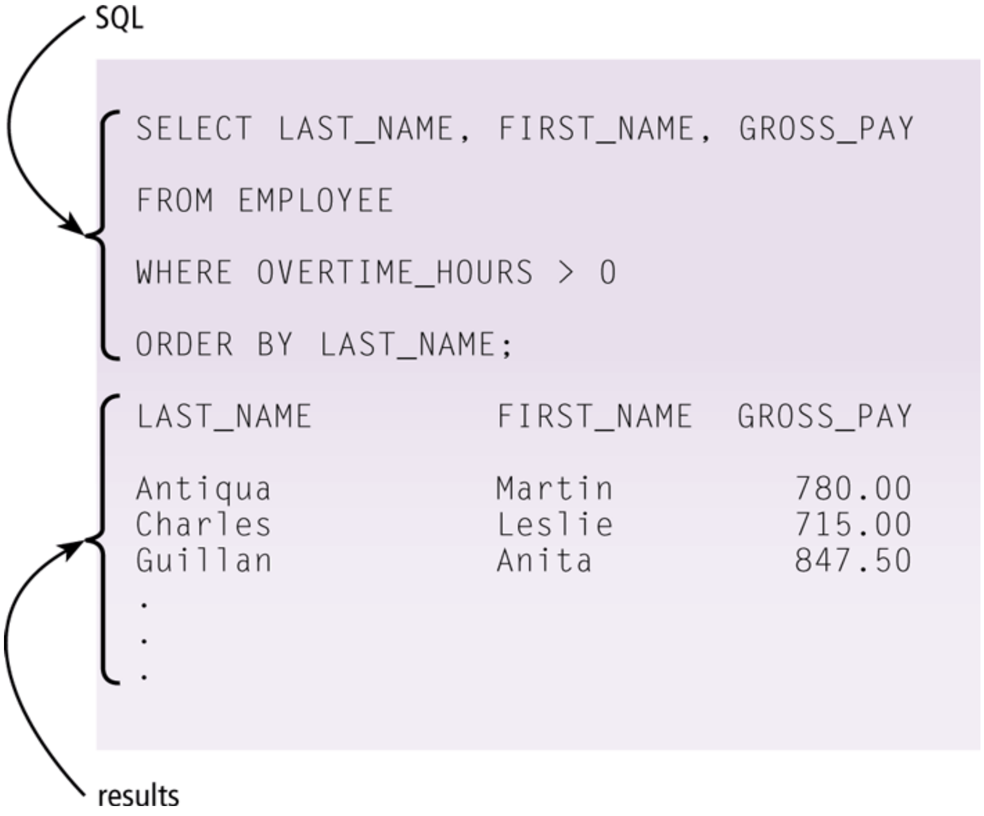
```
int a, b, c;
a = 83;
b = -2;
c = a + b;
```

4GL: Non-procedural Languages

- Very High-Level Languages
- Programmers define only what they want the computer to do, without supplying all the details of how it has to be done.
- The code is written in English-like sentences.
  - Example of a 4GL is the query language that allows a user to request information from a database.
- **Advantages:**
  - Code is easier to maintain
  - Enhances the productivity of the programmers as they have to type fewer lines of code to get something done.
  - A minimum of training by both programmers and non-programmers
- **Disadvantages:**
  - Do not make efficient use of machine's resources.

4GL: Non-procedural Languages

- **Examples of Non-Procedural Languages:**
  - SQL, QBE, Intellect,…


SQL
results

```
SELECT LAST_NAME, FIRST_NAME, GROSS_PAY
FROM EMPLOYEE
WHERE OVERTIME_HOURS > 0
ORDER BY LAST_NAME;
```

```
LAST_NAME           FIRST_NAME   GROSS_PAY

Antiqua             Martin          780.00
Charles             Leslie          715.00
Guillan             Anita           847.50
.
.
.
```

5GL: Non-procedural Languages

- 5GLs are centered on solving problems using constraints given to the program, rather than using an algorithm written by a programmer.
- They are widely used in artificial intelligence research.
- **Knowledge-based languages.**
- Examples of a 5GL: Prolog, OPS5, and Mercury.
- Resembles to the "natural“ language.

Choosing a Language

- The decision on which language to use is dependent:
  - Work environment
  - Suitability for the task
  - Speed of development or speed of execution
  - Expertise of the programmer
  - …

3. Introduction to Python

- Python is a programming language that is easy to learn and very powerful.
  - It is free and well documented.
  - Runs everywhere.
  - A clean syntax.

    → Used in many universities for introductory courses.
  - Well supported by tools.
  - Main language used for web programming at Google.
  - Widely used in scientific computation, for instance at NASA, by mathematicians and physicists.
  - Large portions of games (such as Civilization IV) are written in Python.

The History of Python

History of Python

Python was created by Guido van Rossum in the Netherlands in 1990

Simple, concise, and intuitive syntax and extensive library

A general-purpose programming language

Python code is translated and executed by an interpreter

An object-oriented programming (OOP) language

One of the largest and best-organized open source projects going.

Two versions of Python are : Python 2 and Python 3

Environments

- Install **Anaconda** Environment: allows
  - Terminal running Python Interactively (Anaconda Prompt)
  - Running Scripts (Spyder IDE)
  - Jupyter Notebook
- **Anaconda**
  - A package of goodies
  - Widely used for data analysis.
  - Many useful Python libraries preinstalled
  - Versions for major Operating Systems (OS)

Q&A