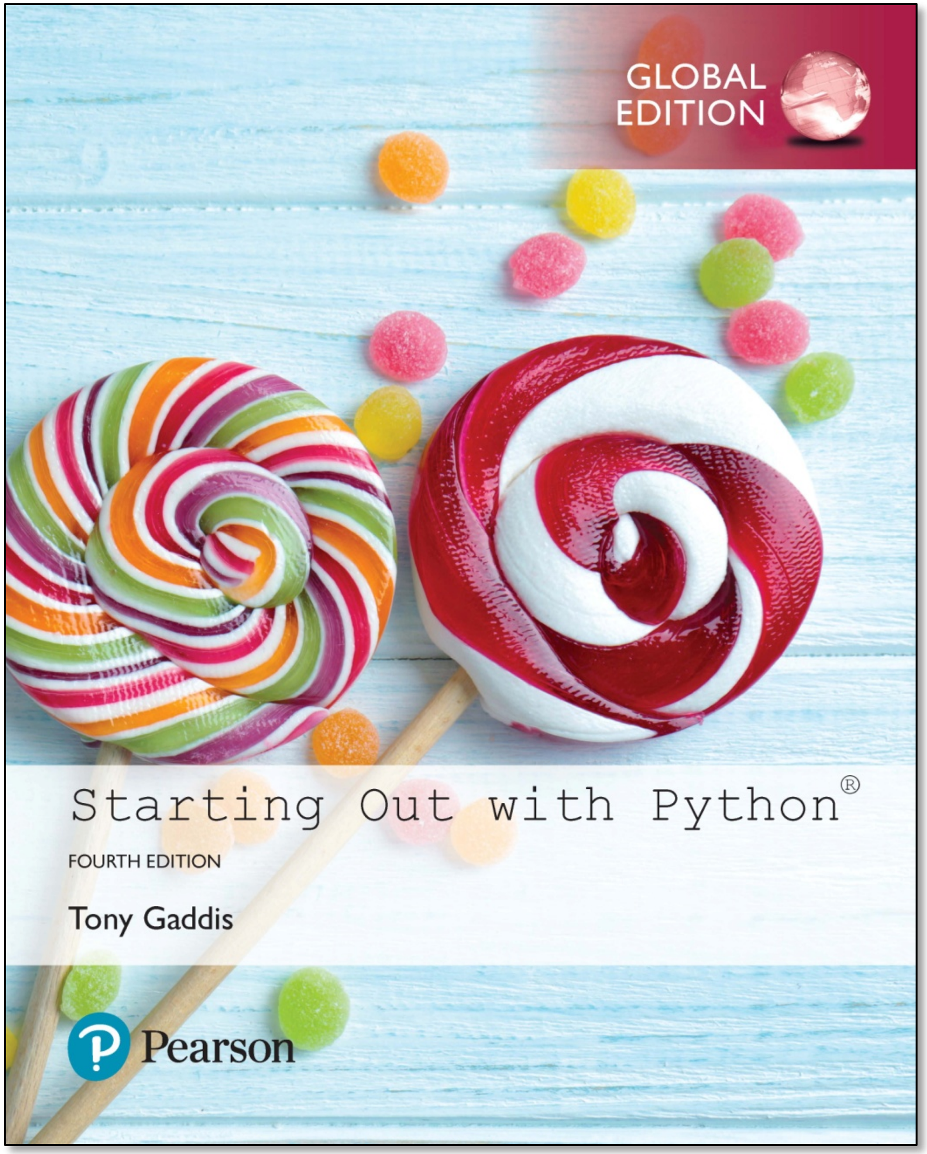
CHAPTER 1

Introduction to Computers and Programming



Topics

- Introduction
- Hardware and Software
- How Computers Store Data
- How a Program Works
- Using Python

Introduction - What is Computer?

Computer is a device that can perform computations and make logical decisions billions of times faster than human beings can.

Many of today's personal computers can perform several billion additions per second.

Computer Technology and Us!

Nowadays, **almost everything** around us is somehow influenced by computer technology. We are interacting with computer technology almost every stages in our everyday life.

- Playing Games on Computers or on other devices
- GSM networks and communication systems
- Traffic lights, speeding radars, and security cameras
- Online processing (paying bills, bank transactions, or registering for courses).
- Elevators or TV systems
- Most importantly, challenging engineering problems are solved by using computer technology and computer programs.

All of these much or less uses computer technology and requires computer programming.

Introduction – What is a Program?

- **Computers are programmable devices**
  - Designed to do any job that a program tells them to
- **Program: set of instructions given to a computer to perform a specific task**
  - Commonly referred to as *Software*
- **Programmer: person who can design, create, and test computer programs**
  - Also known as software developer

Overview of Computers

Computers used/owned by a single person are called Personal Computers (PC). Most commonly used PCs are shown below

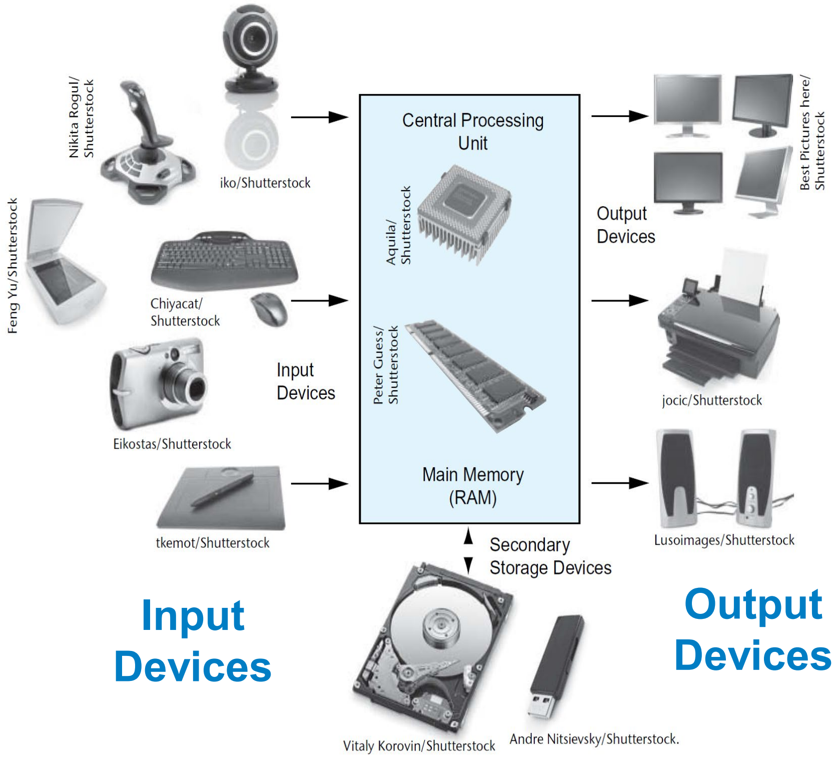
Hardware and Software

- **<u>Hardware</u>: The physical devices that make up a computer**

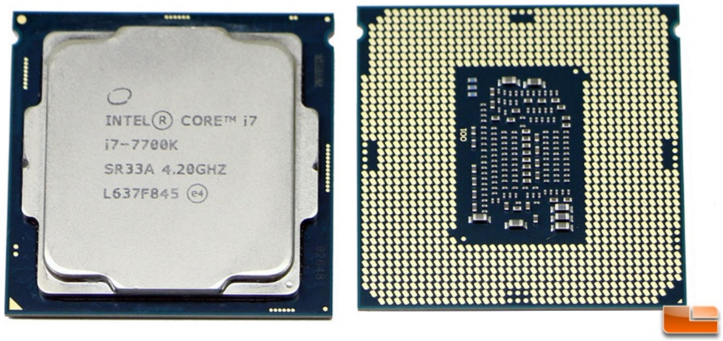
  In other word, it is the electronic part of the computer.

  - Computer is a system composed of several components that all work together

- **Typical major components:**
  - Central processing unit
  - Main memory
  - Secondary storage devices
  - Input and output devices

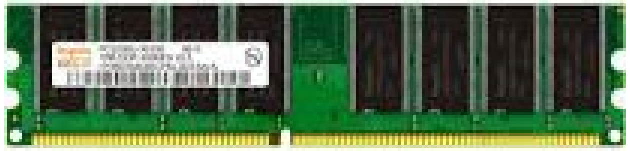


The CPU



- **<u>Central processing unit (CPU)</u>: the part of the computer that actually runs programs**
  - Most important component
  - Without it, cannot run software
  - Used to be a huge device
- **<u>Microprocessors</u>: CPUs located on small chips**

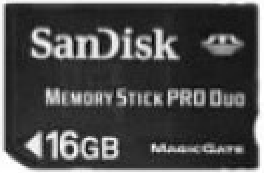
Main Memory



- Main memory: where computer stores a program while program is running, and data used by the program
- Known as *Random Access Memory* or *RAM*
  - CPU is able to quickly access data in RAM
  - Volatile memory used for temporary storage while program is running
  - Contents are erased when computer is off

Secondary Storage Devices

- **<u>Secondary storage</u>: can hold data for long periods of time – permanent storage**
  - Programs normally stored here and loaded to main memory when needed
- **Types of secondary memory**
  - Disk Drive: magnetically encodes data onto a spinning circular disk
  - Optical devices: data encoded optically
  - Flash memory: portable, no physical disk
  - Solid state drive: faster than disk drive, no moving parts, stores data in solid state memory



Input Devices

- **Input: data the computer collects from people and other devices**
- **Input device: component that collects the data**
  - Examples: keyboard, mouse, touchscreen, scanner, camera
  - Disk drives can be also considered as input devices because they load programs into the main memory

Output Devices

- **Output: data produced by the computer for other people or devices**
  - Can be text, image, audio, or bit stream
- **Output device: formats and presents output**
  - Examples: video display, printer
  - Disk drives and USB drives can be considered output devices because data is sent to them to be saved

Software

- **<u>Software:</u> Program part of the computer**
- **Everything the computer does is controlled by software**
  - General categories:
    - **<u>Application software:</u>** Programs that make computer useful for every day tasks
    - **<u>System software:</u>** Programs that controls the operation of the hardware components

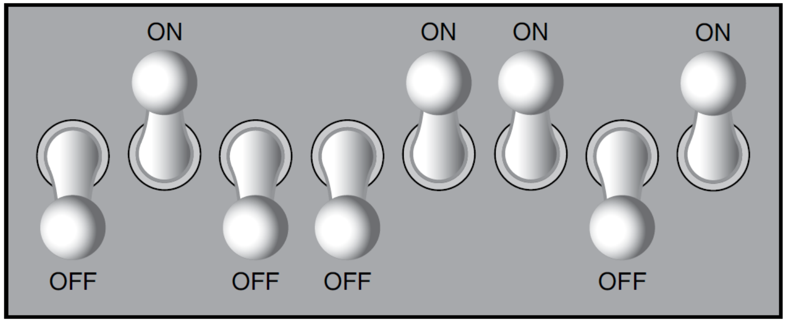
Application Software

System Software

How Computers Store Data

- **All data/information in a computer is stored in sequences of 0s and 1s**
- **Byte: just enough memory to store letter or small number**
  - Divided into eight bits – 1 Byte = 8 bits
  - Bit: electrical component that can hold positive or negative charge, like on/off switch
  - The on/off pattern of 8 bits in a byte represents data stored in the byte

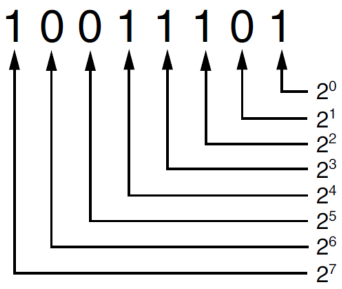
Figure 1-6 Think of a byte as eight switches

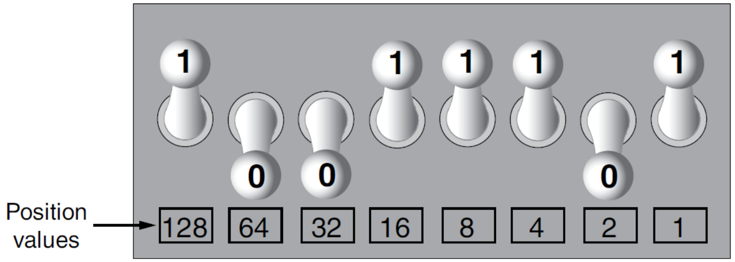


- You may watch video from https://www.youtube.com/watch?v=Xpk67YzOn5w

Storing Numbers

- **Bit represents two values, 0 and 1**
- **Computers use binary numbering system**
  - Position of digit `j` is assigned the value 2^{j-1}
  - To determine value of binary number sum position values of the 1s
  - For example: 10011101 corresponds in 157.



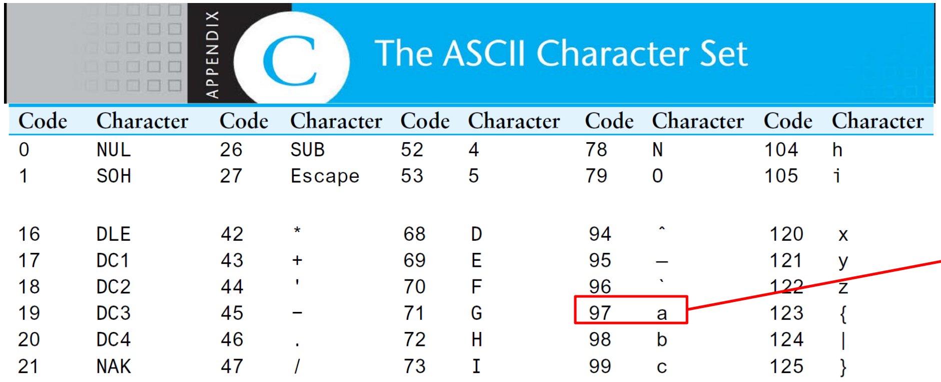


128 + 16 + 8 + 4 + 1 = **157**

- **Byte size limits are 0 and 255**
  - 0 = all bits off; 255 = all bits on
  - To store larger number, use several bytes

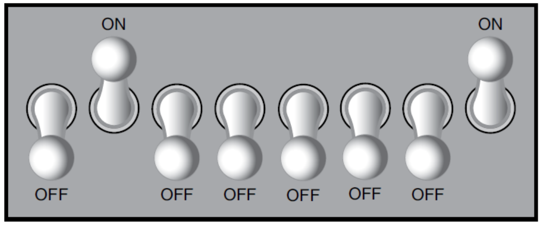
Storing Characters

- **Data stored in computer must be stored as binary number**
- **Characters are converted to numeric code, numeric code stored in memory**
  - Most important coding scheme is ASCII
    - ASCII is limited: defines codes for only 128 characters – *See Appendix C for ASCII List.*
  - Unicode coding scheme becoming standard
    - Unicode is superset of ASCII
    - Can represent characters for other languages



APPENDIX C The ASCII Character Set

Code	Character	Code	Character	Code	Character	Code	Character	Code	Character
0	NUL	26	SUB	52	4	78	N	104	h
1	SOH	27	Escape	53	5	79	O	105	i
16	DLE	42	*	68	D	94	ˆ	120	x
17	DC1	43	+	69	E	95	—	121	y
18	DC2	44	'	70	F	96	`	122	z
19	DC3	45	–	71	G	97	a	123	{
20	DC4	46	.	72	H	98	b	124	|
21	NAK	47	/	73	I	99	c	125	}



The letter A stored in a byte.

Advanced Number Storage

- **To store negative numbers and real numbers, computers use binary numbering and encoding schemes**
  - Negative numbers encoded using two’s complement

You may watch the video from https://www.youtube.com/watch?v=tCMjsQU3_TM

  - Real numbers encoded using floating-point notation

Other Types of Data

- **Digital: describes any device that stores data as binary numbers**
- **Digital images are composed of pixels**
  - To store images, each pixel is converted to a binary number representing the pixel's color
- **Digital music is composed of sections called samples**
  - To store music, each sample is converted to a binary number

How a Program Works

- **CPU is designed to perform simple operations on pieces of data**
  - Examples: reading data, adding, subtracting, multiplying, and dividing numbers
  - Understands instructions written in machine language and included in its instruction set
    - Each brand of CPU has its own instruction set
- **To carry out meaningful calculation, CPU must perform many operations**
- **Instructions and data to CPU is passed as binary system.**

How a Program Works (cont’d.)

- **Program must be copied from secondary memory to RAM each time CPU executes it**

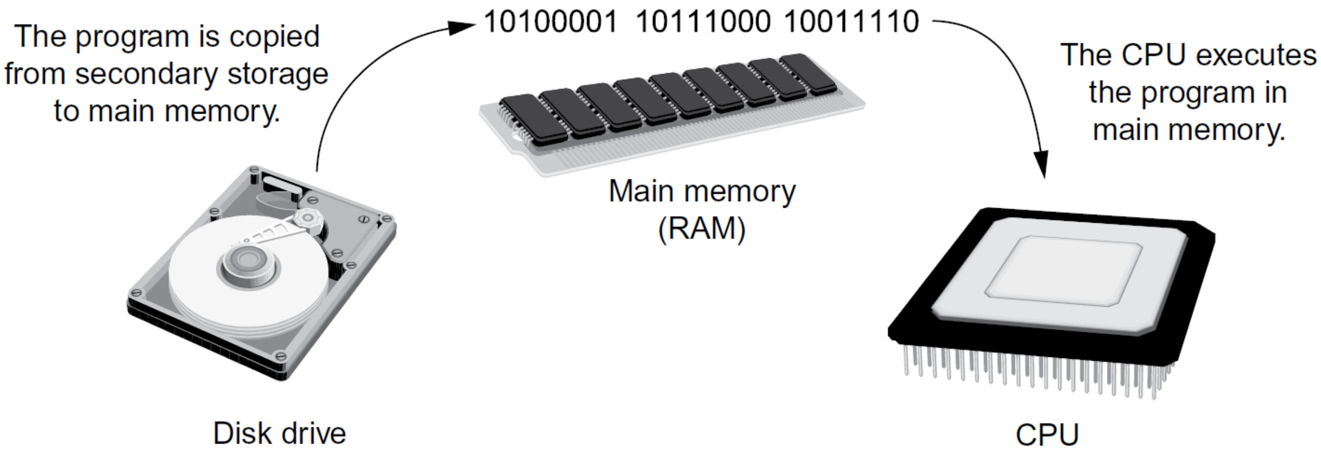

The program is copied from secondary storage to main memory.
10100001 10111000 10011110
The CPU executes the program in main memory.
Main memory (RAM)
Disk drive
CPU

Figure 1-15 A program is copied into main memory and then executed

How a Program Works (cont'd.)

- **CPU executes program in cycle:**

  Fetch: read the next instruction from memory into CPU

  Decode: CPU decodes fetched instruction to determine which operation to perform

  Execute: perform the operation

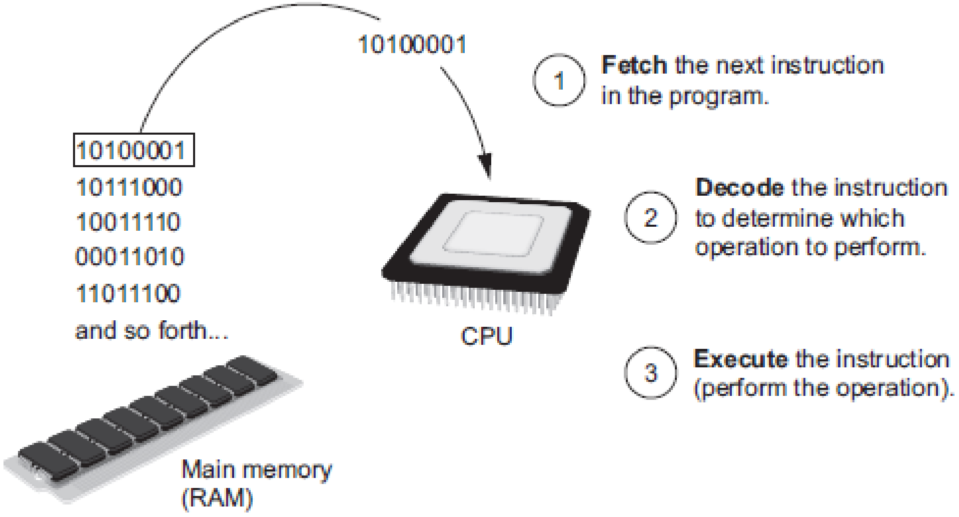

10100001
10100001
10111000
10011110
00011010
11011100
and so forth...
Main memory (RAM)
CPU
1 Fetch the next instruction in the program.
2 Decode the instruction to determine which operation to perform.
3 Execute the instruction (perform the operation).

Figure 1-16 The fetch-decode-execute cycle

How to write a Program?

Remember computer program is set of instructions given to computer to get something done.

How can we give instructions to a computer?

In order to instruct computer, we must instruct in the way that a computer can understand. So we need to speak/talk the language that the computer understand, this is why the programming languages exist.

There are two types of programming languages

- Low-Level Languages (LLL)
- High-Level Languages (HLL)

The LLL and HLL have advantages and disadvantages.

Low-Level and High-Level Languages

- **Low-level language: close in nature to machine language**
  - Not easy to learn
  - Instructions given directly to CPU in terms of 1s and 0s
  - CPU dependent – so not much preferred
  - Example: assembly language
- **High-Level language: allows simple creation of powerful and complex programs**
  - Easy to learn – more understandable
  - No need to know how CPU works or write large number of instructions – no worries about 1s and 0s
  - Instructions are close to real language (mostly English)
  - CPU independent

-Terminology in HLL-
Key Words, Operators, Syntax and Statement

- **Key words: predefined words used to write program in high-level language**
  - Each key word has specific meaning: print , input etc.
- **Operators: perform operations on data**
  - Example: math operators to perform arithmetic: +, - etc.
- **Syntax: set of rules to be followed when writing program**
- **Statement: individual instruction used in high-level language.** *For example* print(“Hello”) *is a statement.*

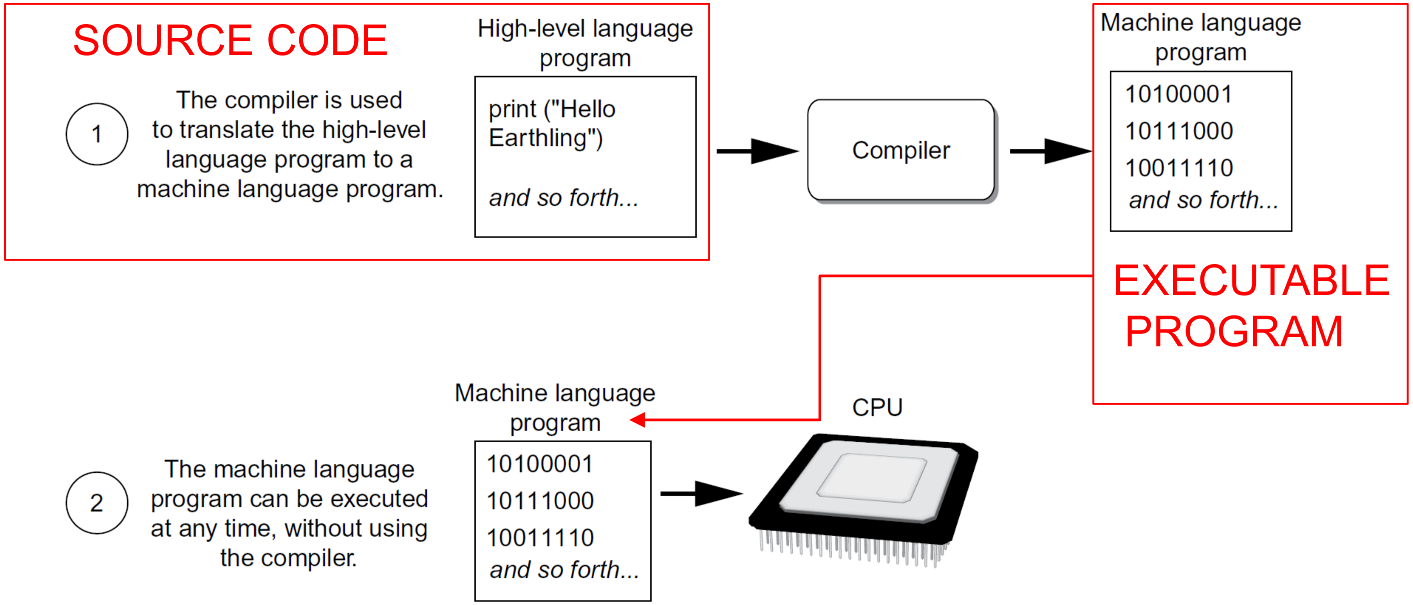
Translation of the HLL Program into Machine Language Compilers and Interpreters

- **Programs written in high-level languages must be translated into machine language to be executed**

- **The program is then translated into machine language by using**
  - Compiler
  - Interpreter

Compilers

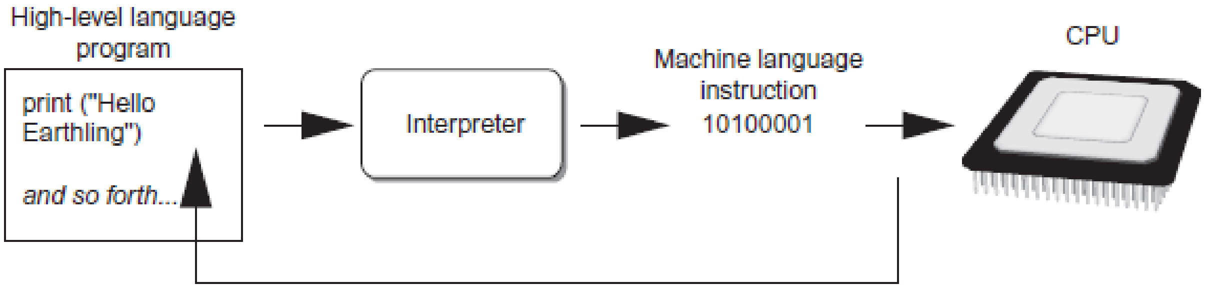
- **Compiler: translates high-level language program into separate machine language program**
  - Machine language program can be executed at any time
  - setup.exe and all exe programs are compiled programs.


SOURCE CODE
1
The compiler is used to translate the high-level language program to a machine language program.
High-level language program
print ("Hello Earthling")
and so forth...
Compiler
Machine language program
10100001
10111000
10011110
and so forth...
EXECUTABLE PROGRAM
2
The machine language program can be executed at any time, without using the compiler.
Machine language program
10100001
10111000
10011110
and so forth...
CPU

- Syntax error: prevents code from being compiled!

Interpreters

- **<u>Interpreter</u>: translates and executes instructions in high-level language program**
  - Used by Python language
  - Interprets one instruction at a time
  - No separate machine language program


High-level language program
print ("Hello Earthling")
and so forth...
Interpreter
Machine language instruction
10100001
CPU

The interpreter translates each high-level instruction to its equivalent machine language instructions and immediately executes them.

This process is repeated for each high-level instruction.

- **<u>Source code</u>: statements written by programmer**
- **<u>Syntax error</u>: prevents code from being compiled!**

Using Python

- **Python must be installed and configured prior to use**
  - One of the items installed is the Python interpreter
  - You may obtain the Python https://www.python.org/downloads/

- **Python interpreter can be used in two modes:**
  - **Interactive mode:** enter statements on keyboard
  - **Script mode:** save statements in Python script

Student may also see also Section 1.5 from the textbook.

Interactive Mode

- **Open the terminal and type python to start Python in interactive mode**
- **When you start Python in interactive mode, you will see a prompt**
  - Indicates the interpreter is waiting for a Python statement to be typed
  - Prompt reappears after previous statement is executed
  - Error message displayed If you incorrectly type a statement
  - Ctrl + Z is used to exit the Python
- **Good way to learn new parts of Python**

Writing Python Programs and Running Them in Script Mode

- **Statements entered in interactive mode are not saved as a program**
- **To have a program use script mode**
  - Save* a set of Python statements in a file
  - The filename should have the **.py** extension
  - To run the file, or script, type

    ```
    python filename
    ```

    at the operating system command line

**You may use Notepad or any other editor to write in your source code.*

The IDLE Programming Environment

- **IDLE (Integrated Development Program): single program that provides tools to write, execute and test a program**
  - Automatically installed when Python language is installed
  - Runs in interactive mode
  - Has built-in text editor with features designed to help write Python programs
  - By using the built-in text editor, you may run your Python programs in script mode.

Summary

- **This chapter covered:**
  - Main hardware components of the computer
  - Types of software
  - How data is stored in a computer
  - Basic CPU operations and machine language
  - Fetch-decode-execute cycle
  - Complex languages and their translation to machine code
  - Installing Python and the Python interpreter modes