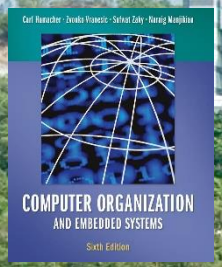
香港中文大學
The Chinese University of Hong Kong

CSCI2510 Computer Organization

Lecture 01: Basic Structure of Computers

Ming-Chang YANG

mcyang@cse.cuhk.edu.hk



Reading: Chap. 1.1~1.3

Outline

- Computer: Tools for the Information Age
- Basic Functional Units of a Computer
  - Input
  - Output
  - Memory
  - Processor
- Basic Operational Concepts
  - Program and Instruction

What are computers used for?

https://www.bhphotovideo.com/explora/computers/buying-guide/holiday-2012-overview-tablet-computers

Computer Types (1/4)

- **Personal Computer**: used by dedicated individual with the support of a variety of applications.
  - Mobile Computer
  - Notebook Computer
  - Desktop Computer
  - Workstation Computer





https://www.titancomputers.com/Titan-X150-Intel-Xeon-E3-1200-V3-Series-Video-Ed-p/x150.htm
https://www.qvcuk.com/Apple-iMac-27%22-5K-Retina-w-Intel-Core-i5-8GB-RAM%2C-1TB-HDD-%26-2yr-Tech-Support.product.508688.html
https://www.amazon.ca/Microsoft-Surface-NVIDIA-GeForce-graphics/dp/B0163GS05Q
https://www.appworldin.com/product/ipad-pro-12-9inch-wifi-cellular-256gb-gold/
https://gadgets.ndtv.com/apple-iphone-x-4258

Computer Types (2/4)

- **Servers and Enterprise Systems**: meant to be shared by a potentially large number of users.

- **Supercomputers**: the most expensive computers used for the highly demanding computations.

https://www.anandtech.com/show/12124/dell-emc-launches-poweredge-xr2-rugged-server-1u-44-cores-512-gb-ram-30-tb-storage
https://www.verdict.co.uk/countries-supercomputers-world/

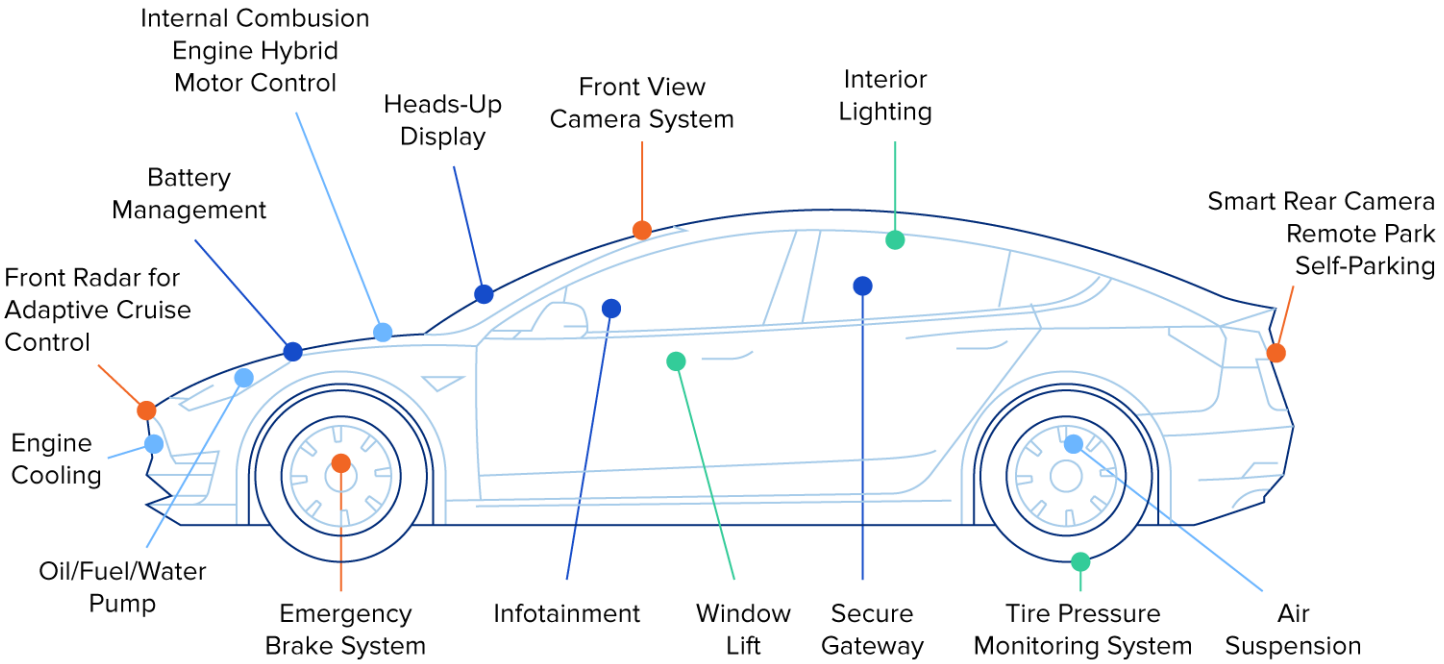
Computer Types (3/4)

- **Grid Computers**: a cost-effective alternative composed of a large number of personal computers in a physically distributed high-speed network.

https://techspirited.com/differences-similarities-between-grid-cluster-computing

Computer Types (4/4)

- **Embedded Computers**: integrated into a device and used for a specific purpose.

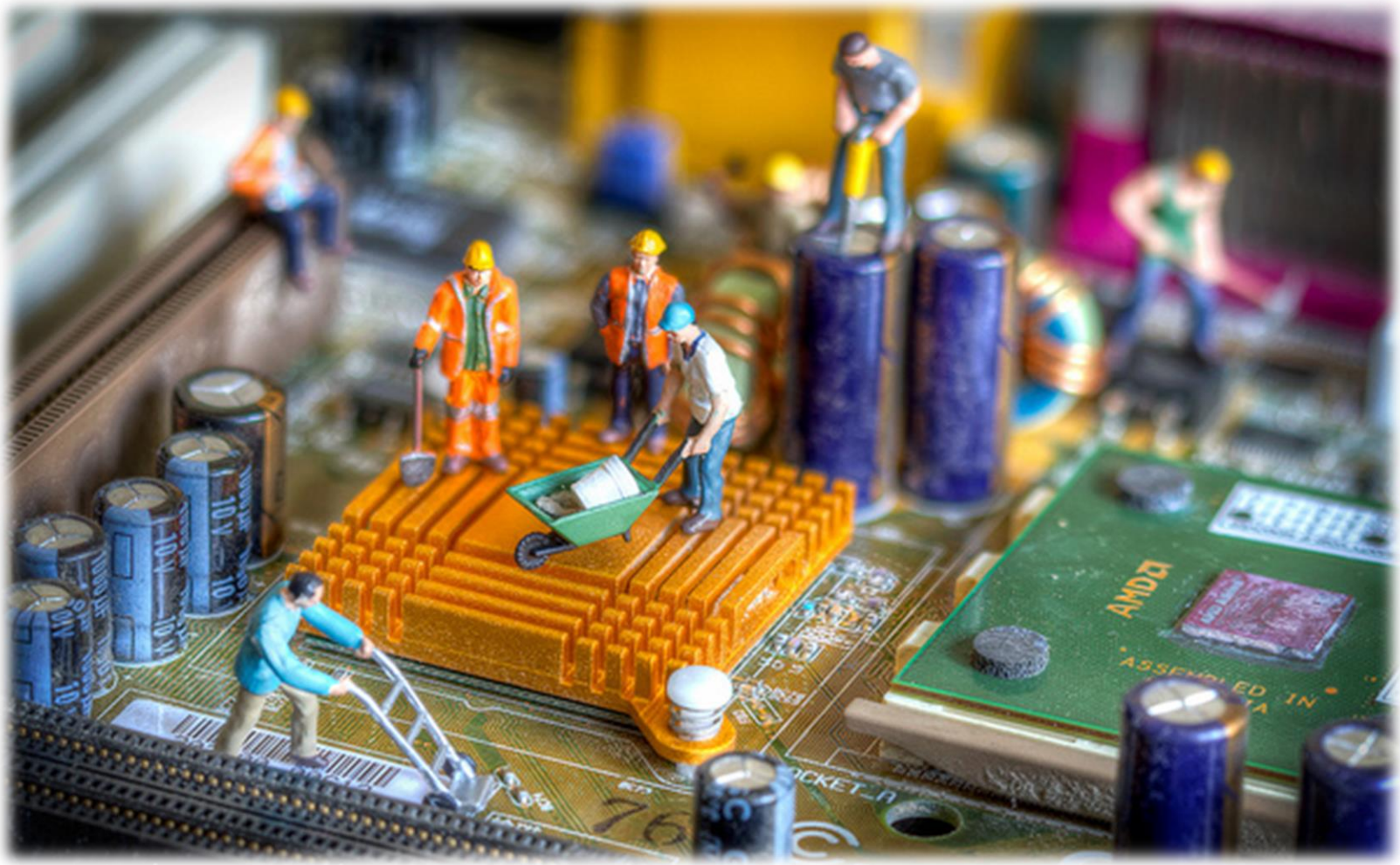


https://www.rs-online.com/designspark/applications-of-embedded-systems-1
https://www.toptal.com/insights/agile-talent/embedded-systems-design-agile-talent

Outline

- Computer: Tools for the Information Age
- Basic Functional Units of a Computer
  - Input
  - Output
  - Memory
  - Processor
- Basic Operational Concepts
  - Program and Instruction

What is inside a computer?



https://itexperts.co.za/8-things-happening-inside-computer-box/

Math Quiz!

- Try to answer the following questions:

Q1. $4 \times 7 + 5 = ?$

(A) 19 (B) 48 (C) 33 (D) 29

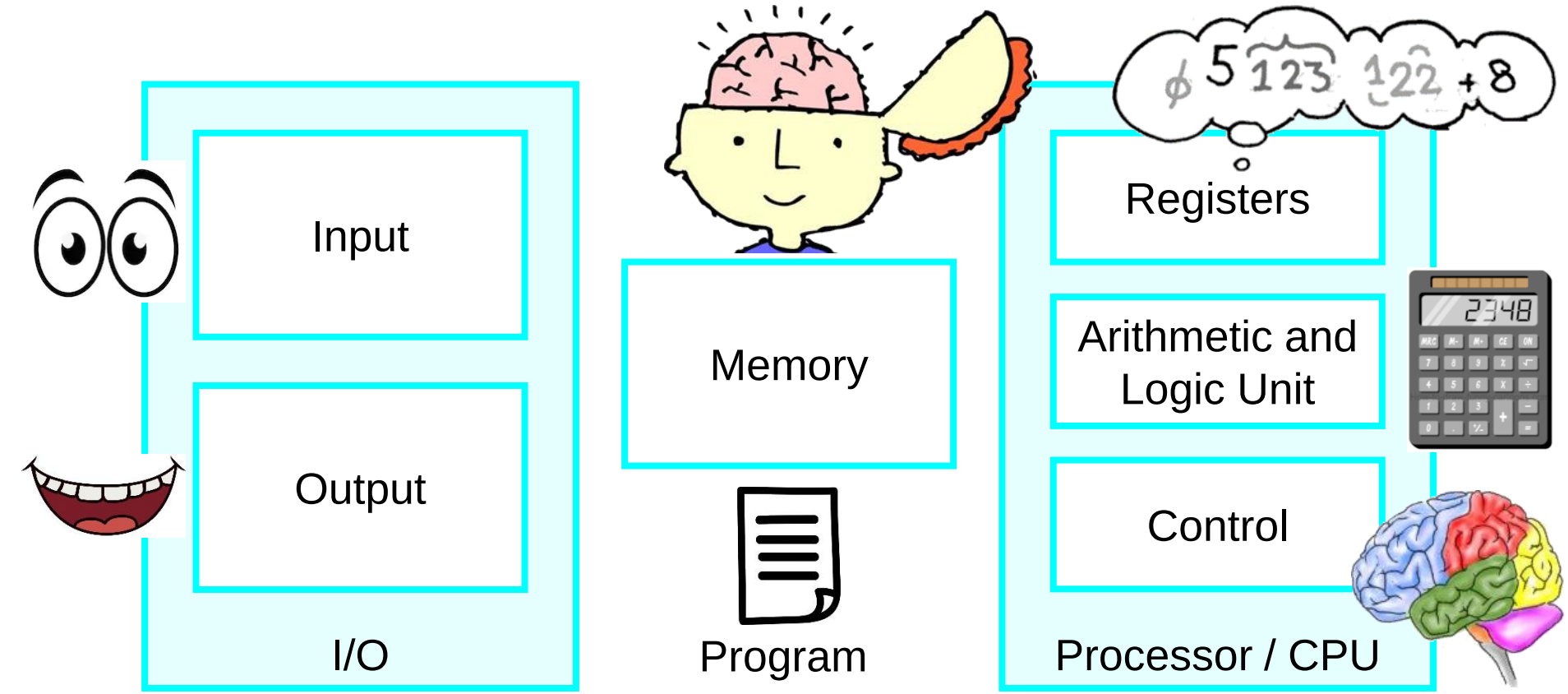
Q2. $3 - 3 \times 2 + 9 = ?$

(A) 3 (B) 6 (C) 9 (D) 12

Q3. $6 + 5 \times 8 - 1 \times (-3) = ?$

(A) 10 (B) 43 (C) 91 (D) 49

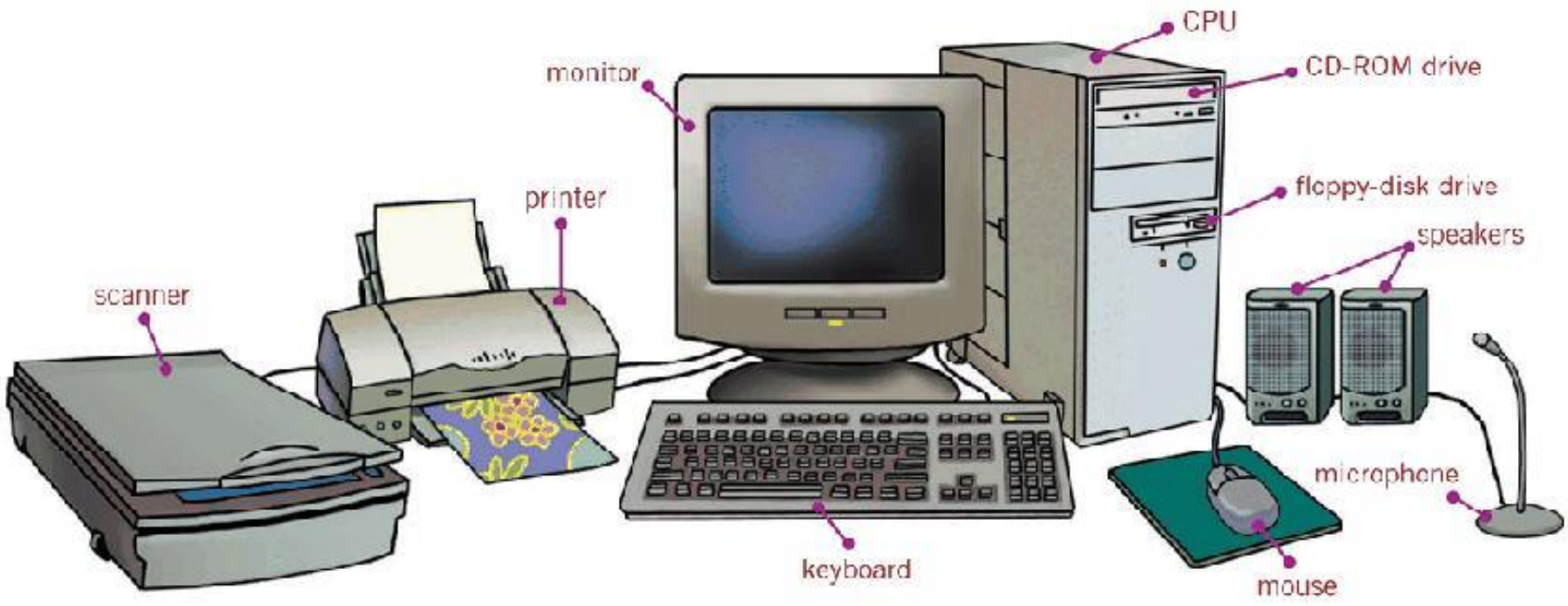
Basic Functional Units of a Computer



- **Input**: accepts coded information from human operators.
- **Memory**: stores the received information for later use.
- **Processor**: executes the instructions of a program stored in the memory.
- **Output**: reacts to the outside world.
- **Control**: coordinates all these actions.

Overview: Input and Output Units

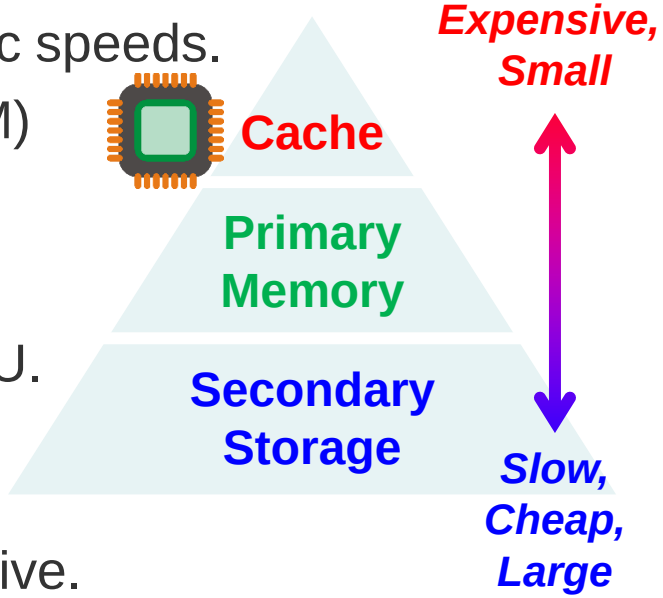
- **Input**: keyboard, mouse, microphone, CDROM, etc.
- **Output**: graphical display, printer, etc.
- The Collective Term: Input/Output (I/O) units.



https://norizman.wordpress.com/notes/

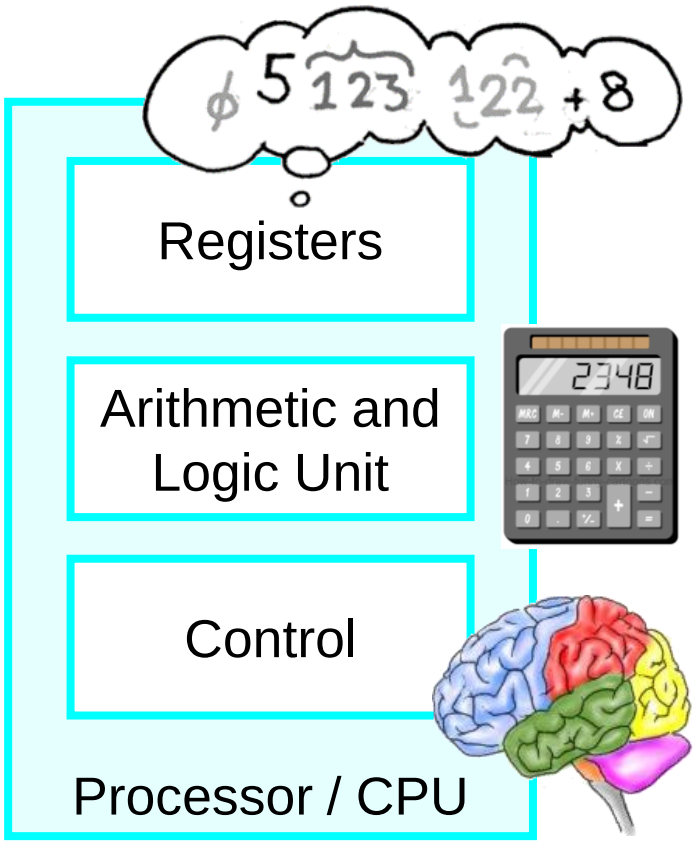
Overview: Memory Unit (Hierarchy)

- **Memory** is used to store programs and data.
- There are two classes of memory/storage:
  - **Primary Memory** (also called **Main Memory**)
    - A fast memory that operates at electronic speeds.
    - Example: random-access memory (RAM)
    - **Cache Memory**: A smaller, faster RAM to hold parts of a program (and data) that are currently being executed by CPU.
  - **Secondary Storage**
    - Primary memory is essential but expensive.
    - Additional, less expensive, permanent secondary storage is used when large amounts of data and many programs have to be stored.
    - Example: solid-state drive (SSD), hard disk (HDD), CD, DVD, etc.



Overview: Processor Unit

- **Registers**
  - Very small but fast memory for storing intermediate values in a computation (inside the processor)
- **Arithmetic & Logic Unit (ALU)**
  - Perform computations
    - Arithmetic Operations: add, subtract, multiply, divide, etc.
    - Logical Operations: and, or, not, etc.
    - Operands are stored in registers.
- **Control Unit**
  - Control the transfer of data and sequencing of operations among memory, registers, ALU, I/O, etc.



Class Exercise 1.1

Student ID: _____ Date:
Name: _____ _____

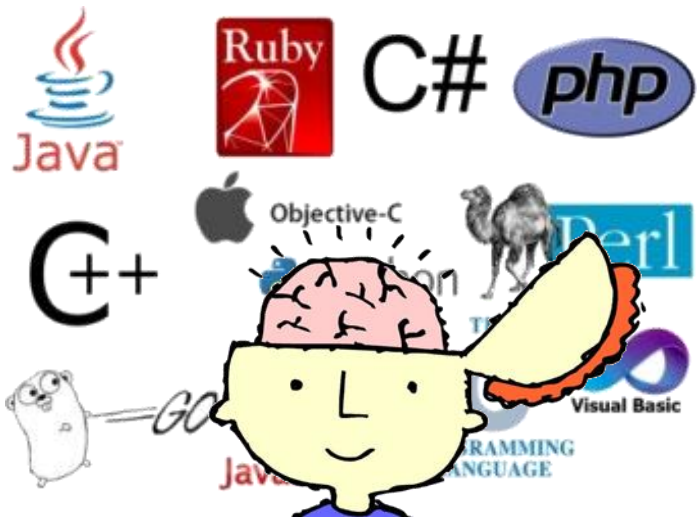
- Question: Fill in the blanks by specifying the corresponding "unit" of a computer (i.e., input, output, memory, registers, arithmetic and logic unit, control).

Math question (e.g., $4 \times 7 + 5$)	
Arithmetic rules (e.g., $\times$ before $+$), or Multiplication table (e.g., $4 \times 7 = 28$)	
Temporary sum (e.g., $4 \times 7 = 28$)	
Computation (e.g., $28 + 5 = 33$)	
Execute rules (e.g., when to read input, when to compute and stop, etc.)	
Answer to the question (e.g., (C) 33)	

Outline

- Computer: Tools for the Information Age
- Basic Functional Units of a Computer
  - Input
  - Output
  - Memory
  - Processor
- Basic Operational Concepts
  - Program and Instruction

How to talk to the computer?



High-level Language

Machine Language

Easy for programmer to understand

Human understandable English words

Language Translation

The computer's own language

Binary numbers (All 1s and 0s)

Example of Language Translation

High-level Language

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
TEMP = V(k);
V(k) = V(k+1);
V(k+1) = TEMP;
```

C/Java Compiler

Fortran Compiler

Assembly Language

```
lw $0, 0($2)
lw $1, 4($2)
sw $1, 0($2)
sw $0, 4($2)
```

instructions!

MIPS Assembler

`lw:` loads a word from **memory** into a register
`sw:` saves a word from a register into **RAM**
`$0,$1,$2:` registers
`0($2):` treats the value of register $2 + 0 bytes as a location
`4($2):` treats the value of register $2 + 4 bytes as a location

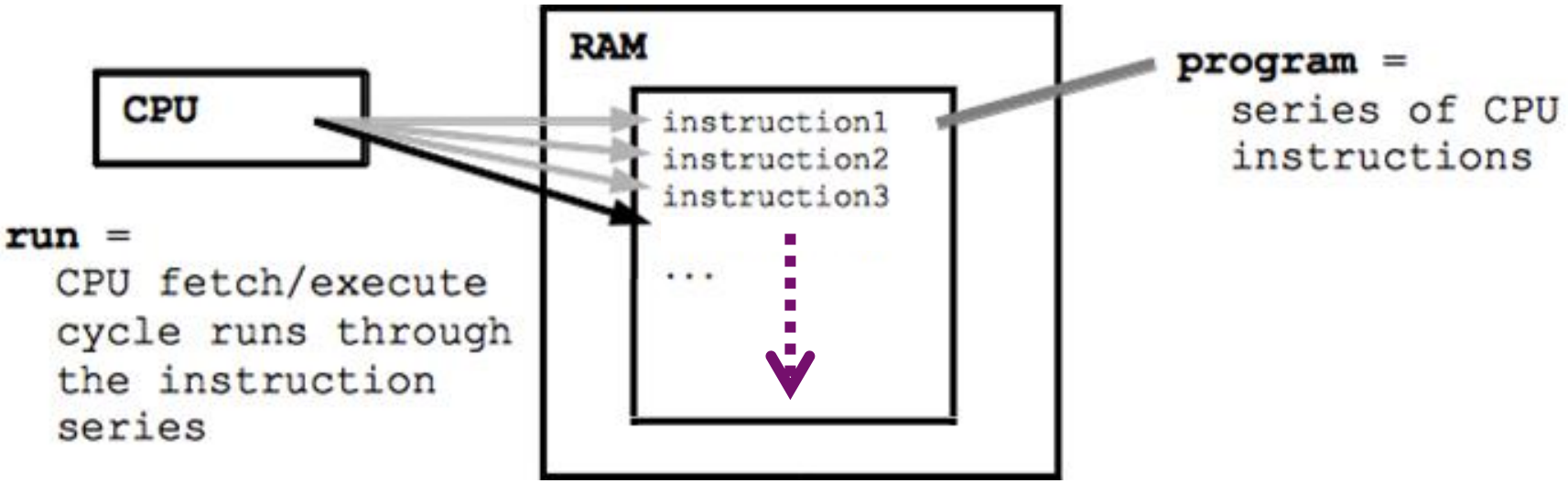
Machine Language

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

https://gerardnico.com/code/lang/machine
https://clip2art.com/explore/Boy%20clipart%20teacher/

Activities in a Computer: Instructions

- A computer is governed by instructions.
  - To perform a given task, a program consisting of a list of machine instructions is stored in the memory.
    - Data to be used as operands are also stored in the memory.
  - Individual instructions are brought from the memory into the processor, one after another, in a sequential way (normally).
  - The processor executes the specified operation/instruction.



An Example of Program Execution

- Considering a program of 3 instructions:

PC → **I_0: Load R0, LOC**

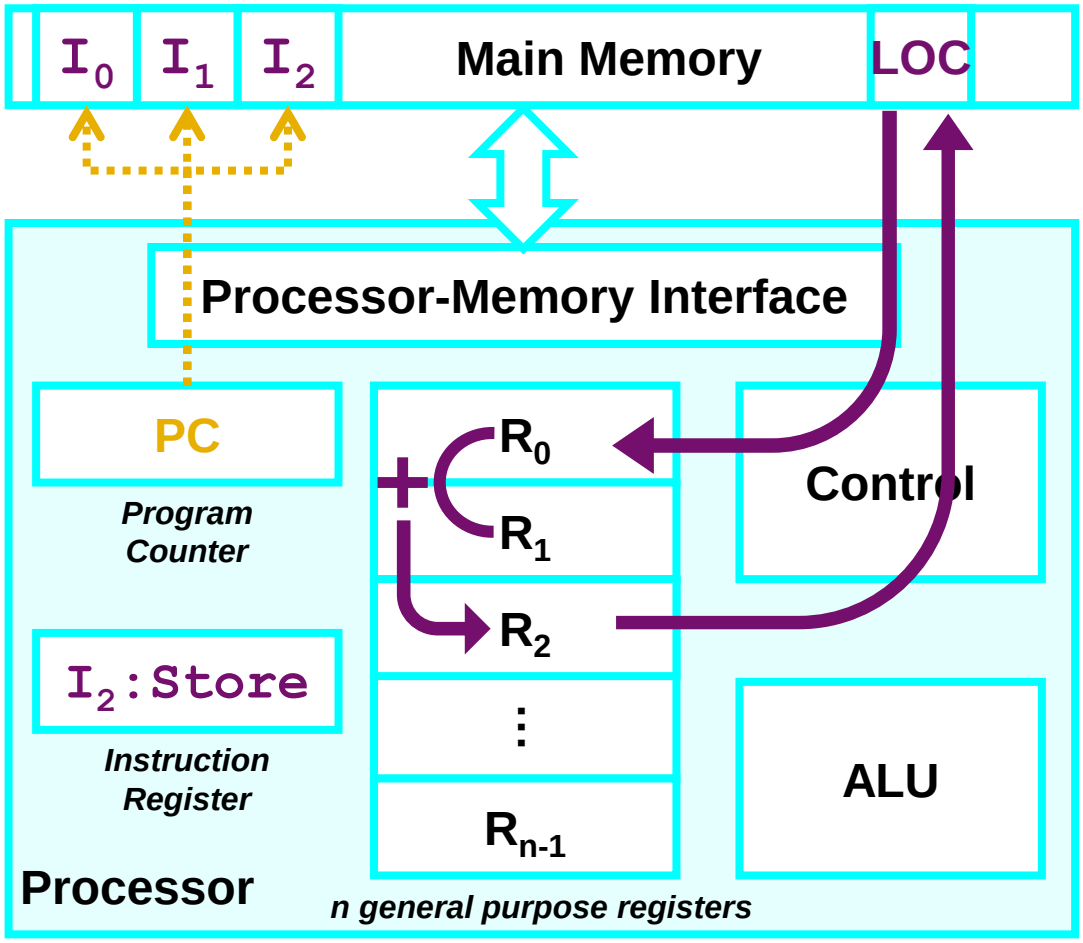
  - Reads the contents of a memory location LOC
  - Loads them into processor register R0

– **I_1: Add R2, R0, R1**

  - Adds the contents of registers R0 and R1
  - Places their sum into register R2

– **I_2: Store R2, LOC**

  - Copies the operand in register R2 to memory location LOC



PC: contains the memory address of the next instruction to be fetched and executed.
IR: holds the instruction that is currently being executed.
R_0~R_{n-1}: n general-purpose registers.

Class Exercise 1.2

- Question: To execute the instruction **`I0: Load R2, LOC`**, what should be the correct order of the following steps?
  A. Send the address of LOC from the instruction in register IR to the memory and issue a Read control command.
  B. Wait until the requested instruction has been retrieved from the memory, then load it into register IR (what be perform is interpreted and determined by the control circuitry).
  C. Send the address of the instruction from register PC to the memory and issue a Read control command.
  D. Wait until the requested word has been retrieved from the memory, then load it into register R2.
  E. Increment the value of register PC to point to the next instruction in memory.
- Answer: _____

Class Exercise 1.3

- Question: Consider the following program, what does this program intend to do?
  - *Hint: Think about (1) use of registers, (2) implementation of the loop, (3) source, destination of operands*
- Answer: _____

LABEL	OPCODE	OPERAND	COMMENT
	`CLEAR`	`R0`	*clear the sum*
	`MOV`	`R2, 10`	*initialize the counter*
`LOOP`	`INPUT`	`A`	*input a new number*
	`ADD`	`R0, A`	*accumulate A into sum*
	`DEC`	`R2`	*subtract one from the counter*
	`JG`	`LOOP`	*conditional branch*
	`MOV`	`SUM, R0`	

Summary

- Computer: Tools for the Information Age
- Basic Functional Units of a Computer
  - Input
  - Output
  - Memory
  - Processor
- Basic Operational Concepts
  - Program and Instruction