



香港中文大學

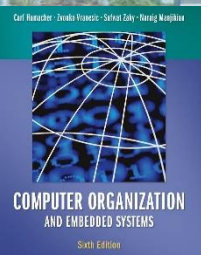
The Chinese University of Hong Kong

# *CSCI2510 Computer Organization*

## **Lecture 01: Basics of Computers**

**Ming-Chang YANG**

[mcyang@cse.cuhk.edu.hk](mailto:mcyang@cse.cuhk.edu.hk)



Reading: Chap. 1.1~1.3



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



# 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**.



# 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.



# Computer Types (4/4)



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



Industrial Robots



GPS Receivers



Digital Cameras



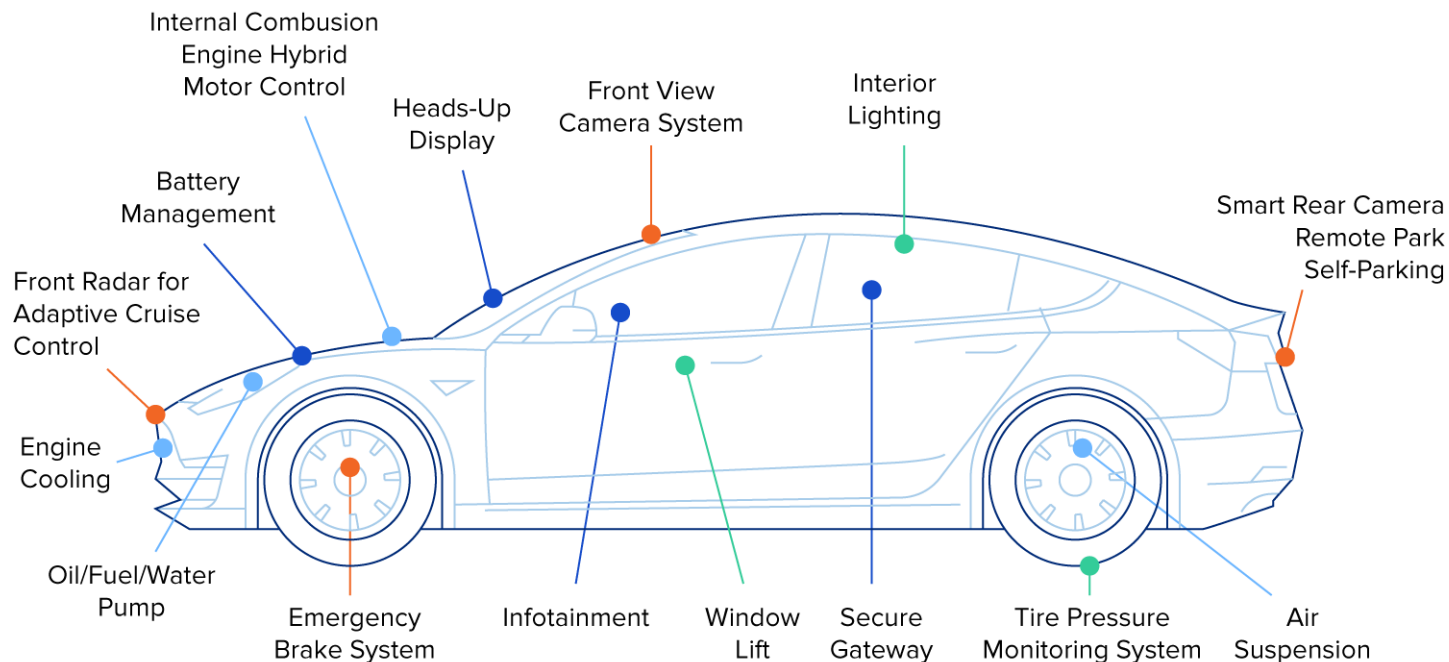
Set top Boxes



Gaming Consoles



Photocopiers

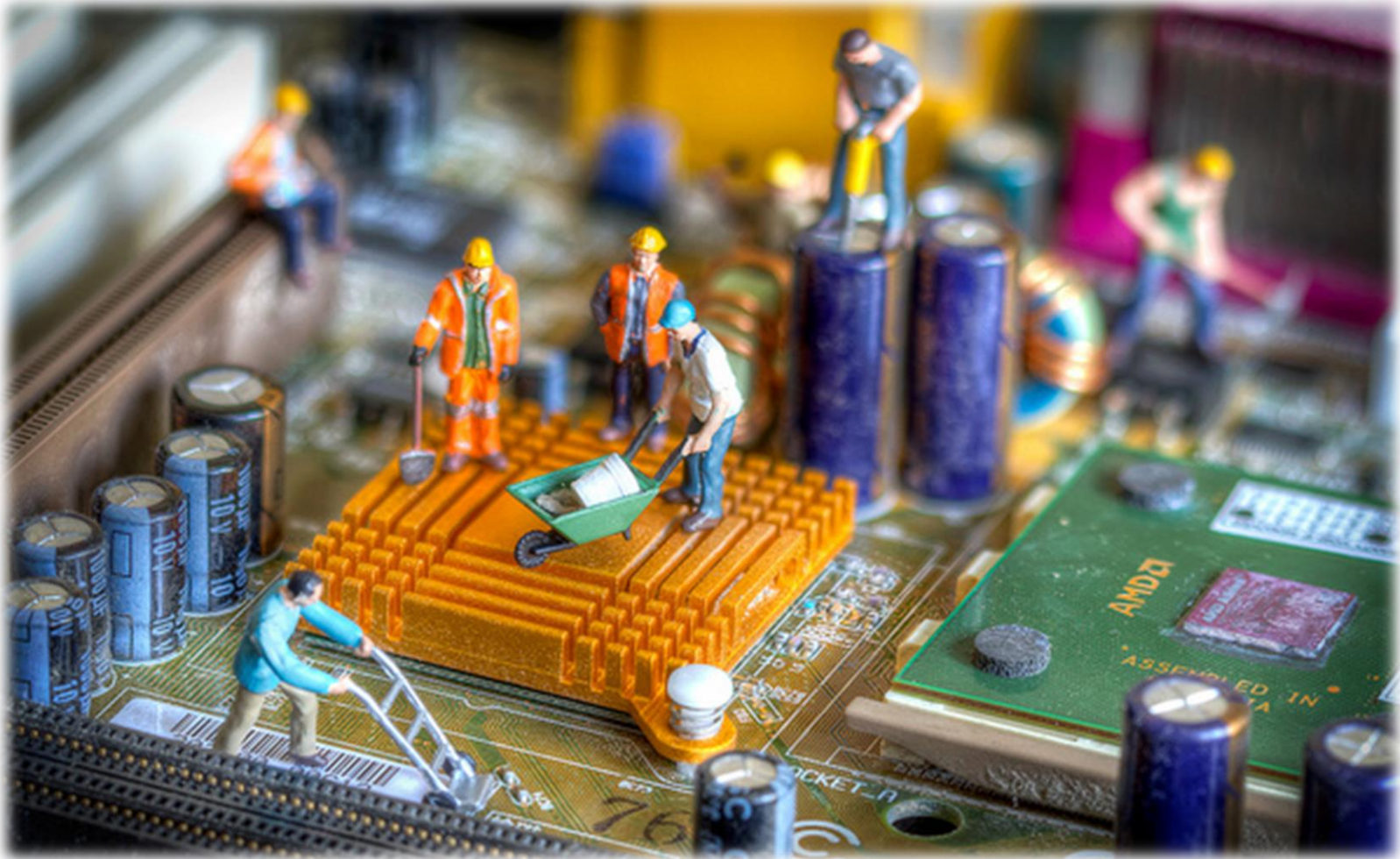




- Computer: Tools for the Information Age
- **Basic Functional Units of a Computer**
  - Input
  - Output
  - Memory
  - Processor
- Basic Operational Concepts
  - Program and Instruction



# What is inside a computer?



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

# Math Quiz!



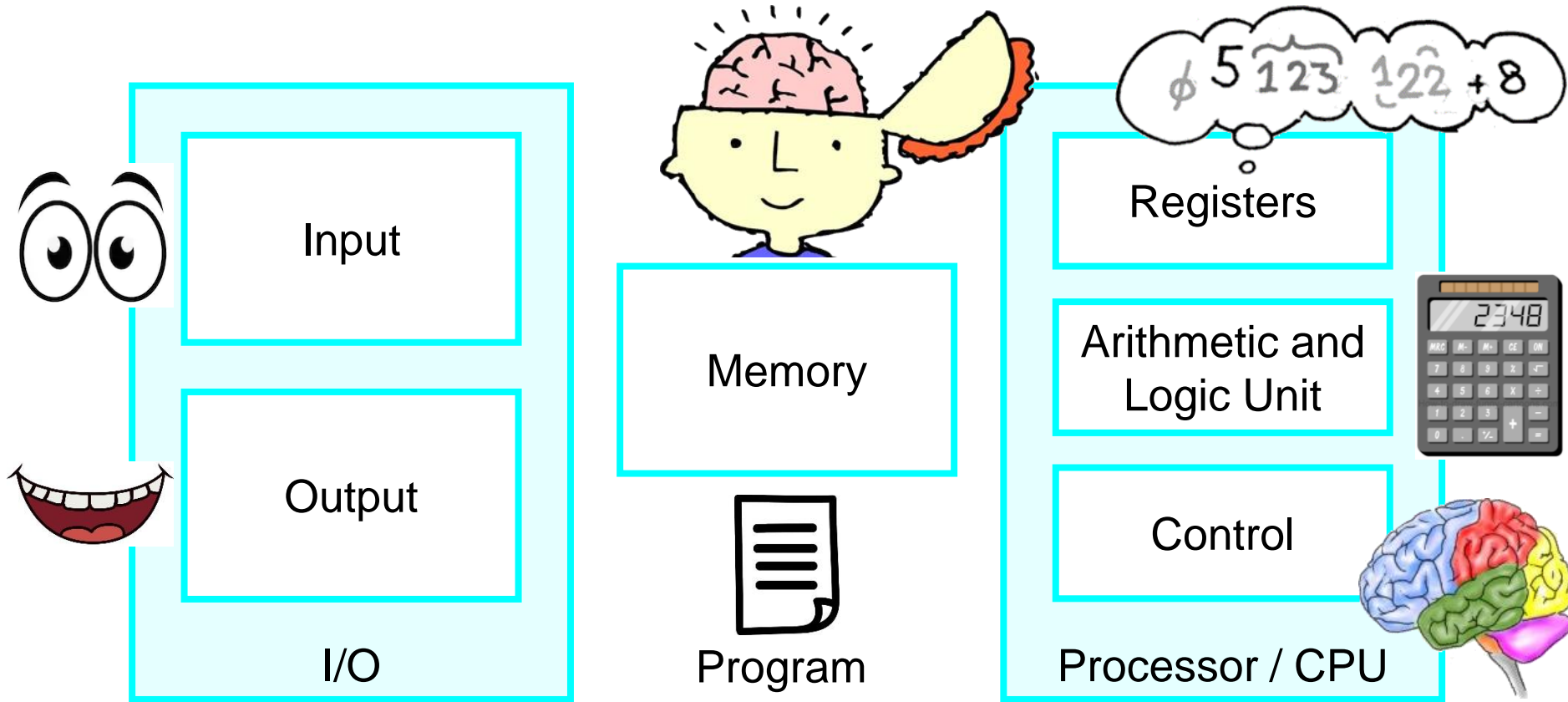
- Try to answer the following math question:

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

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



# Basic Functional Units of a Computer

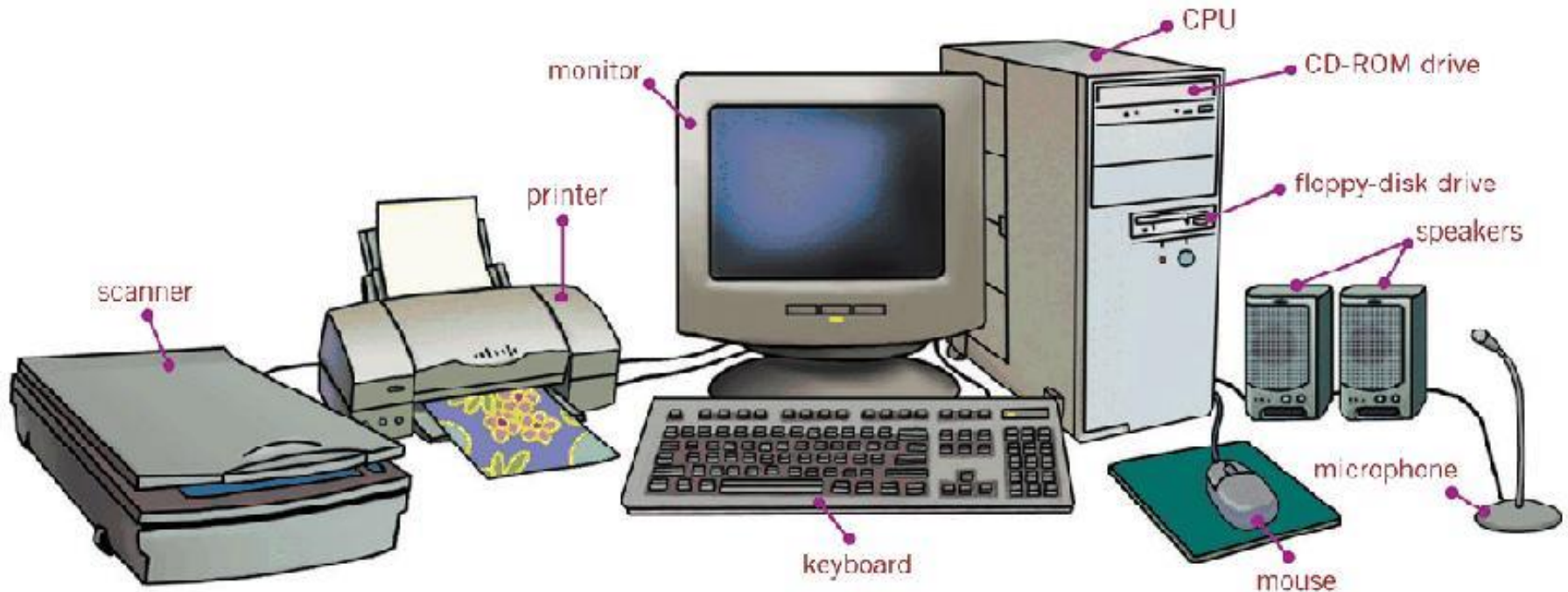


- **Input:** **accepts** coded information from human operators.
- **Memory:** **stores** the received or other important information for later use.
- **Processor:** **executes** the instructions of a program stored in the memory.
- **Output:** **reacts** to the outside world.

# Overview: Input and Output Units



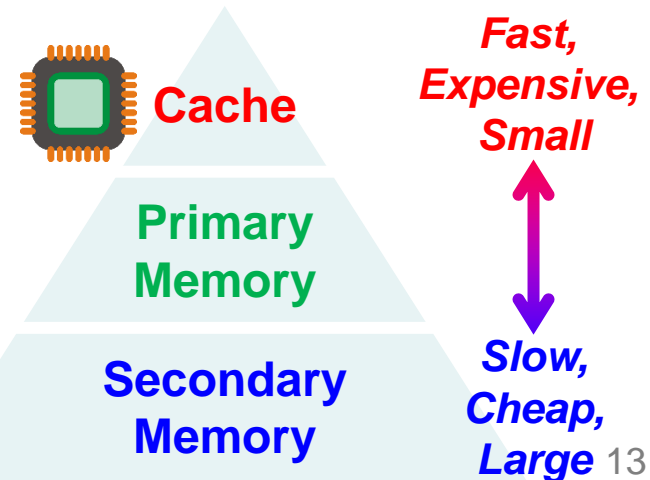
- **Input:** keyboard, mouse, microphone, CDROM, etc.
- **Output:** graphical display, printer, etc.
- *We typically use the collective term, **input/output (I/O)**, to refer to these equipment.*



# Overview: Memory Unit (Hierarchy)



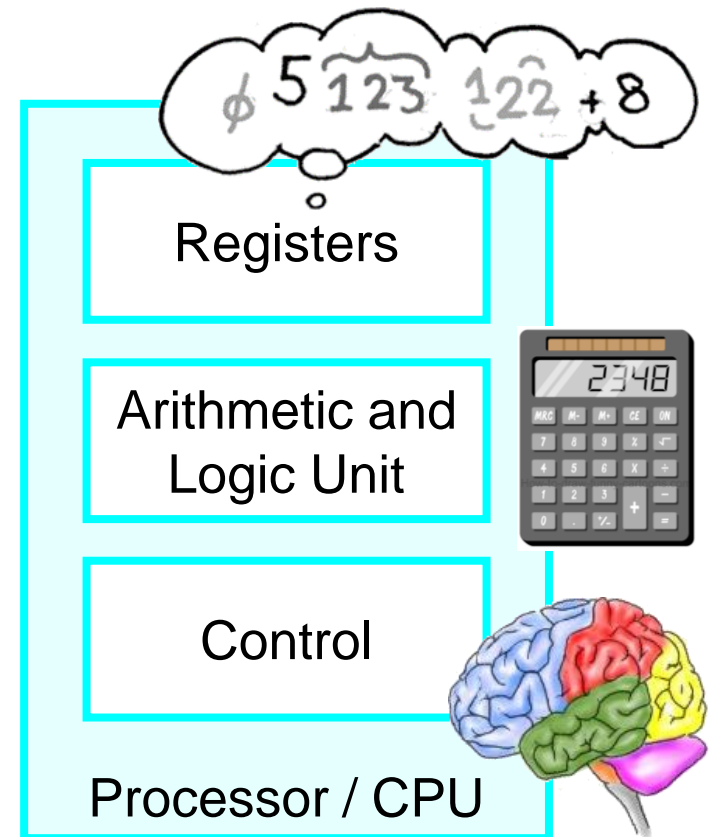
- **Memory** is used to store **programs** and **data**.
- The computer memory is organized as **hierarchies**:
  - **Cache Memory** (pronounced like “cash” \$\$\$)
    - Even smaller but faster memory that can hold parts of a program (and data) being executed by CPU currently.
    - Example: *Static Random-Access Memory* (SRAM)
  - **Primary Memory** (aka **Main Memory**)
    - Small but fast memory that can be operated at electronic speeds.
    - Example: *Dynamic Random-Access Memory* (DRAM)
  - **Secondary Memory** (aka **Storage**)
    - Less expensive but slower memory that can **permanently** store a large amount of program/data.
    - Example: *Solid-State Drive* (SSD), *Hard Disk Drive* (HDD), CD, DVD, etc.



# Overview: Processor Unit



- **Registers**
  - Very small but extreme 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: \_\_\_\_\_

- Fill in the blanks by specifying the corresponding “unit” of a computer (i.e., input, output, memory, registers, arithmetic and logic unit (ALU), control) involved in answering the math quiz.

Math question (e.g.,  $4 \times 7 + 5$ ) ①

Arithmetic rules (e.g.,  $\times$  before  $+$ ), or  
Multiplication table (e.g.,  $4 \times 7 = 28$ ) ②

Temporary sum (e.g.,  $4 \times 7 = 28$ ) ③

Computation (e.g.,  $28 + 5 = 33$ ) ④

Execute rules (e.g., when to read input,  
when to compute and stop, etc.) ⑤

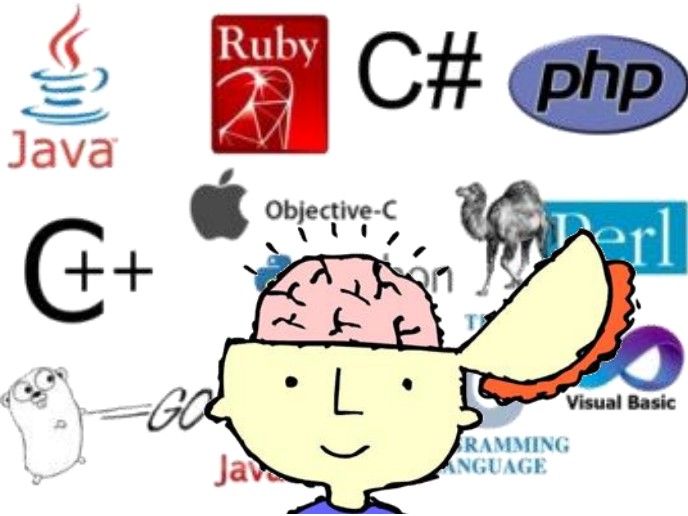
Answer to the question (e.g., (C) 33) ⑥



- 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**

Easy for programmer to understand

Human understandable English words

**Language Translation**

**Machine Language**

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

Machine  
Instructions!

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

Assembly Language

**lw:** load a word from **memory** into a **register**  
**sw:** save a word from a **register** into **memory**  
**\$0, \$1, \$2:** **register identifications**

**0 (\$2):** treat the value of register \$2 + 0 bytes as a location

**4 (\$2):** treat the value of register \$2 + 4 bytes as a location

MIPS Assembler



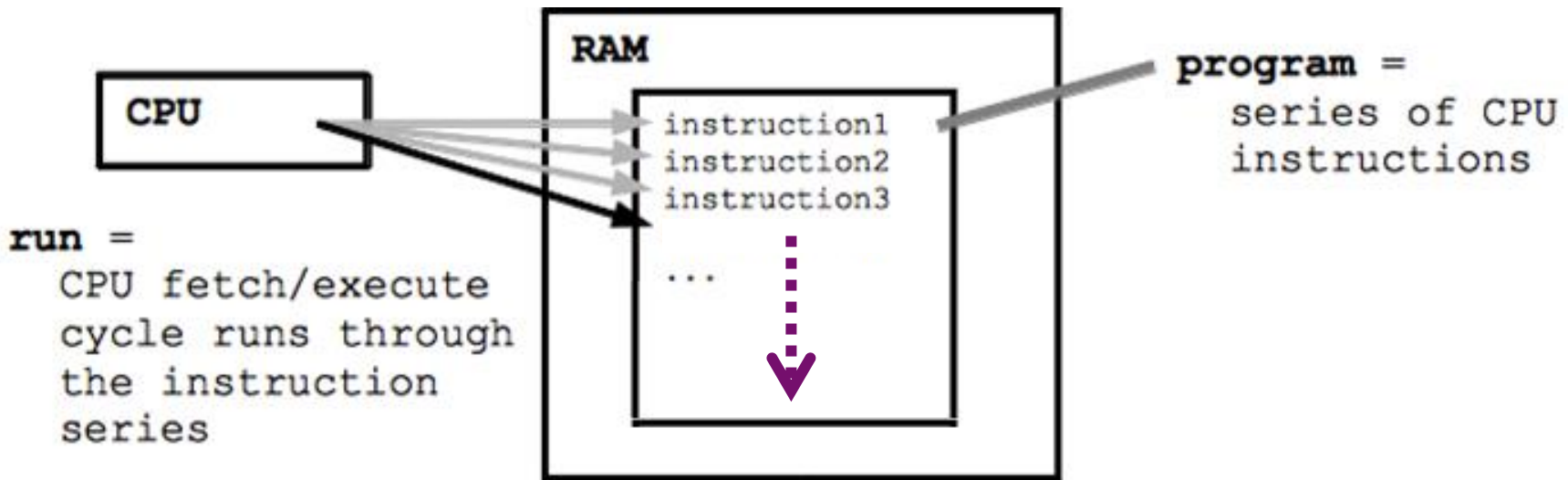
Machine Language

```
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
```

# 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<sub>0</sub>: Load R0, LOC**

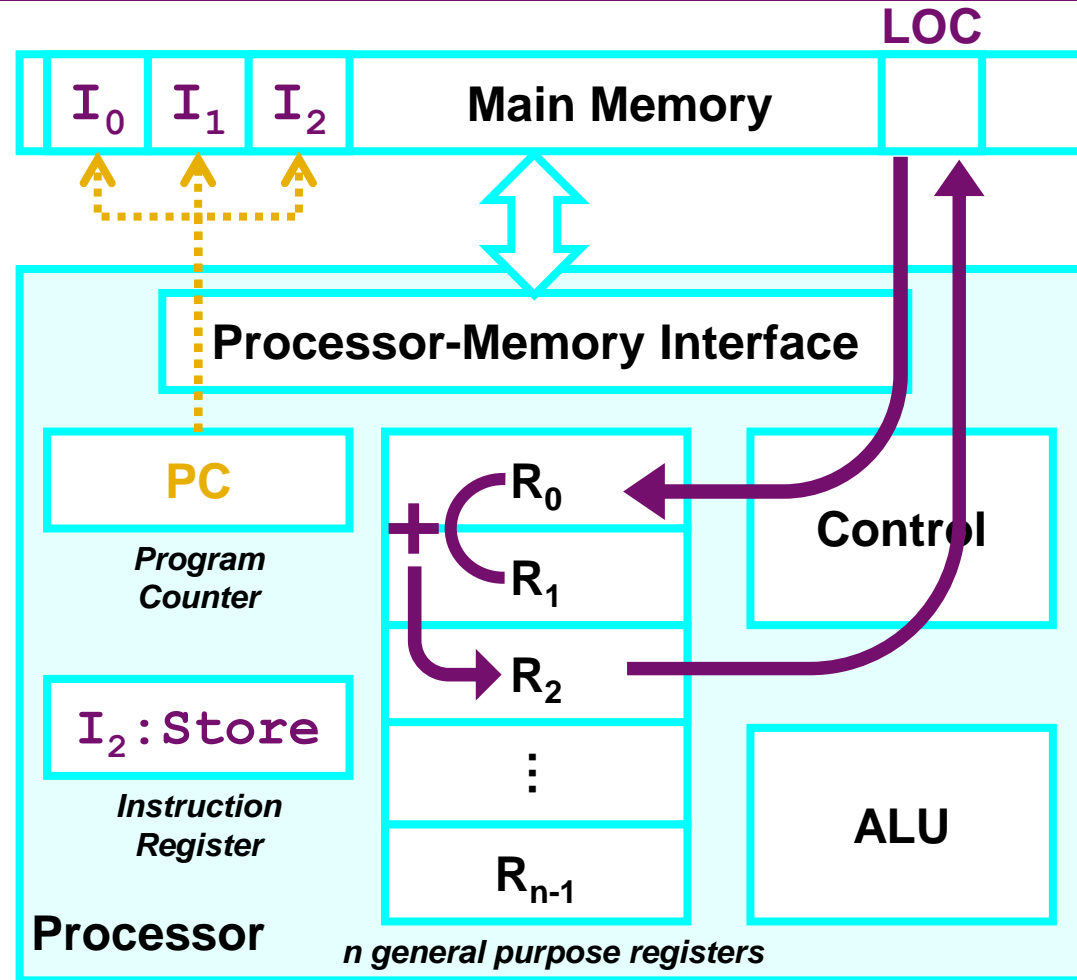
- Reads the contents of a memory location LOC
- Loads them into processor register R0

– **I<sub>1</sub>: Add R2, R0, R1**

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

– **I<sub>2</sub>: 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<sub>0</sub>~R<sub>n-1</sub>**:  $n$  general-purpose registers.

# Class Exercise 1.2



- 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
	<b>CLEAR</b>	R0	
	<b>MOV</b>	R2, 10	
<b>LOOP</b>	<b>INPUT</b>	A	
	<b>ADD</b>	R0, A	
	<b>DEC</b>	R2	
	<b>JG</b>	LOOP	
	<b>MOV</b>	SUM, R0	



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