

Programming Design

| Final Remarks

Ling-Chieh Kung
Department of Information Management
National Taiwan University

Computer programming



- What are **computer programs**?
 - The elements working in computers; also known as **software**.
 - A structured combination of data and instructions used to operate a computer to produce a specific result.
- Strength: High-speed computing, large memory, etc.
- Weakness: People (programmers) need to tell them what to do.
- How may a programmer tell a computer what to do?
 - Programmers use “**programming languages**” to write codes line by line and construct computer programs.
- Running a program means executing the instructions line by line and (hopefully) achieve the programmer’s goal.

Programming languages



- A programming language may be a machine language, an assembly language, or a high-level language (or something else).
- A **machine language** uses 0s and 1s to form instructions.
 - For example, under the MIPS architecture, each instruction is 32-bit long.
 - “00000000001000100011000000100000” means “adding the registers 1 and 2 and placing the result in register 4.”
- An **assembly language** labels these instructions as words.
 - **ADD ax, bx**
MOV cx, ax
 - An **assembler** then translates an assembly program into a machine program.

The C++ programming language



- C++ is developed by Bjarne Stroustrup starting from 1979 at AT&T Bell Labs.
- C++ originates from another programming language C.
 - C is a **procedural** programming language.
 - C++ is an **object-oriented** programming (OOP) language.
- Roughly speaking, C++ is created by adding object-oriented functionalities to C.
 - For **teams** to build **large** software systems requiring a **long** time.
- C++ is (almost) a superset of C.
 - Most C programs can be compiled by a C++ compiler.

The C++ programming language



- C/C++ is sometimes called a “**mid-level**” language.
 - It allows a C++ programmer to “access” the **memory** through pointers.
- With such low-level functionality, C/C++ is powerful.
 - And dangerous...
- In this course, we focus on high-level programming.
 - But talks about memory-level ideas when necessary.
- Who should learn C++?
 - Those who plan to become computer scientists/engineers.
 - Those who want to know all the (system-level) details about a program.
 - Those who want to learn other languages by themselves.

What did we cover?



Week	Date	Lecture	Textbook	Note
1	9/15	Introduction	Chs. 1 & 2	
2	9/22	Selection and repetition	Chs. 3 & 4	
3	9/29	Digital systems	N/A	
4	10/6	Variables and arrays	Ch. 6	
5	10/13	Functions and <i>quiz</i>	Ch. 5	
6	10/20	<i>Midterm exam 1</i>	N/A	
7	10/27	Algorithms and recursion	Chs. 5 & 19	
8	11/3	Time complexity and graph theory	N/A	
9	11/10	Pointers	Ch. 7	
10	11/17	C strings	Ch. 22	
11	11/24	Applications of computer programming	N/A	
12	12/1	<i>Midterm exam 2</i>	N/A	
13	12/8	Applications of computer programming	N/A	
14	12/15	Self-defined data types in C	Ch. 22	
15	12/22	Classes	Chs. 9 & 10	
16	12/29	Operator overloading and C++ Strings	Chs. 8, 11, & 18	
17	1/5	Review and preview	N/A	
18	1/12	<i>Final project presentations</i>	N/A	

Skills that you now have



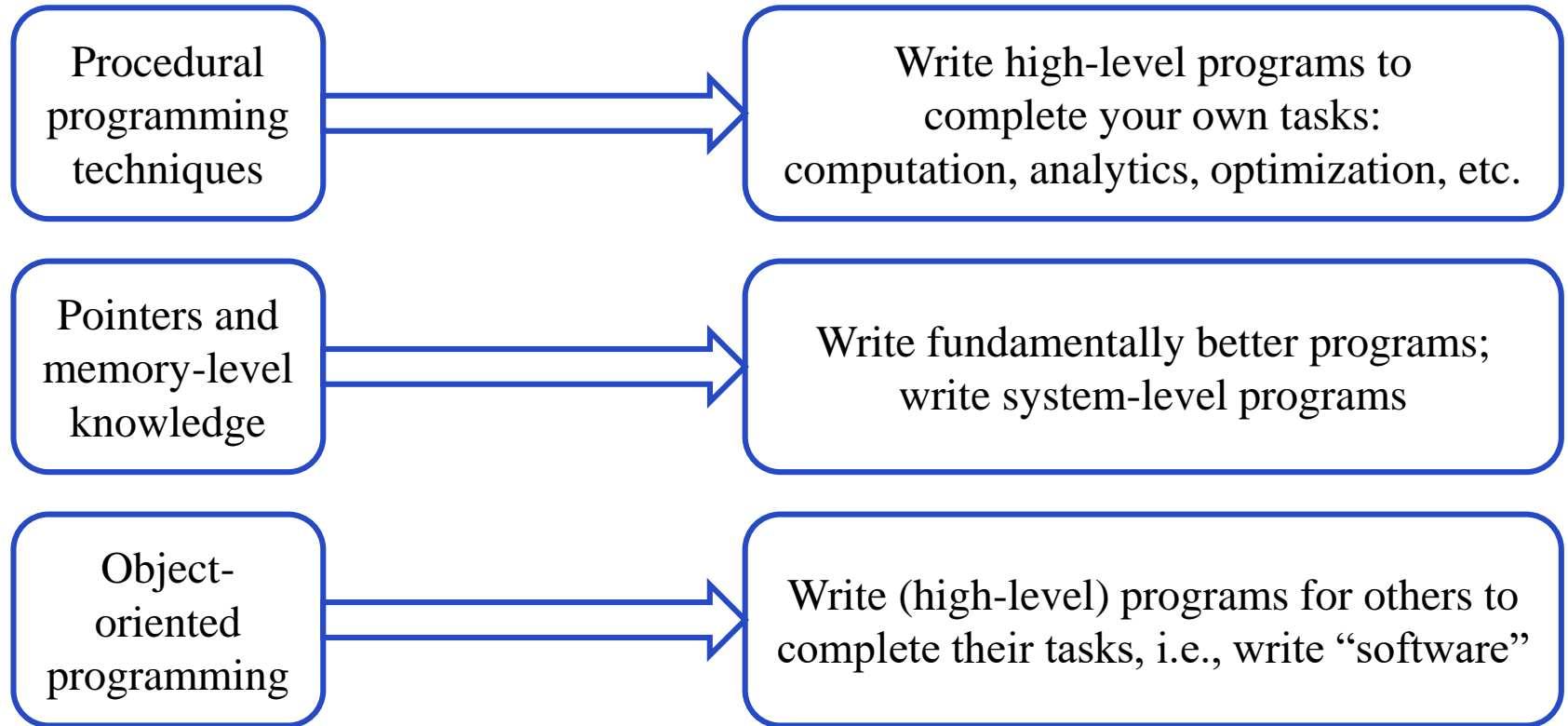
- You know how to use selections, repetitions, arrays, and functions.
 - The most fundamental concepts in computer programming.
- You know randomization, time counting, and string processing.
 - Syntaxes and libraries differ from language to language.
- You know pointers.
 - You know what happens at the memory level.
 - You know how to statically and dynamically allocate memory.
- You know structures and classes.
 - Even operator overloading.

Skills to learn (in the next semester)



- You will learn more about object-oriented programming.
 - C++ strings, file I/O, and header files.
 - Inheritance and polymorphism.
- You will learn some advanced techniques.
 - Templates.
 - Exception handling.
- And of course data structures.
- That is why we teach you C++.

Things that you will be able to do



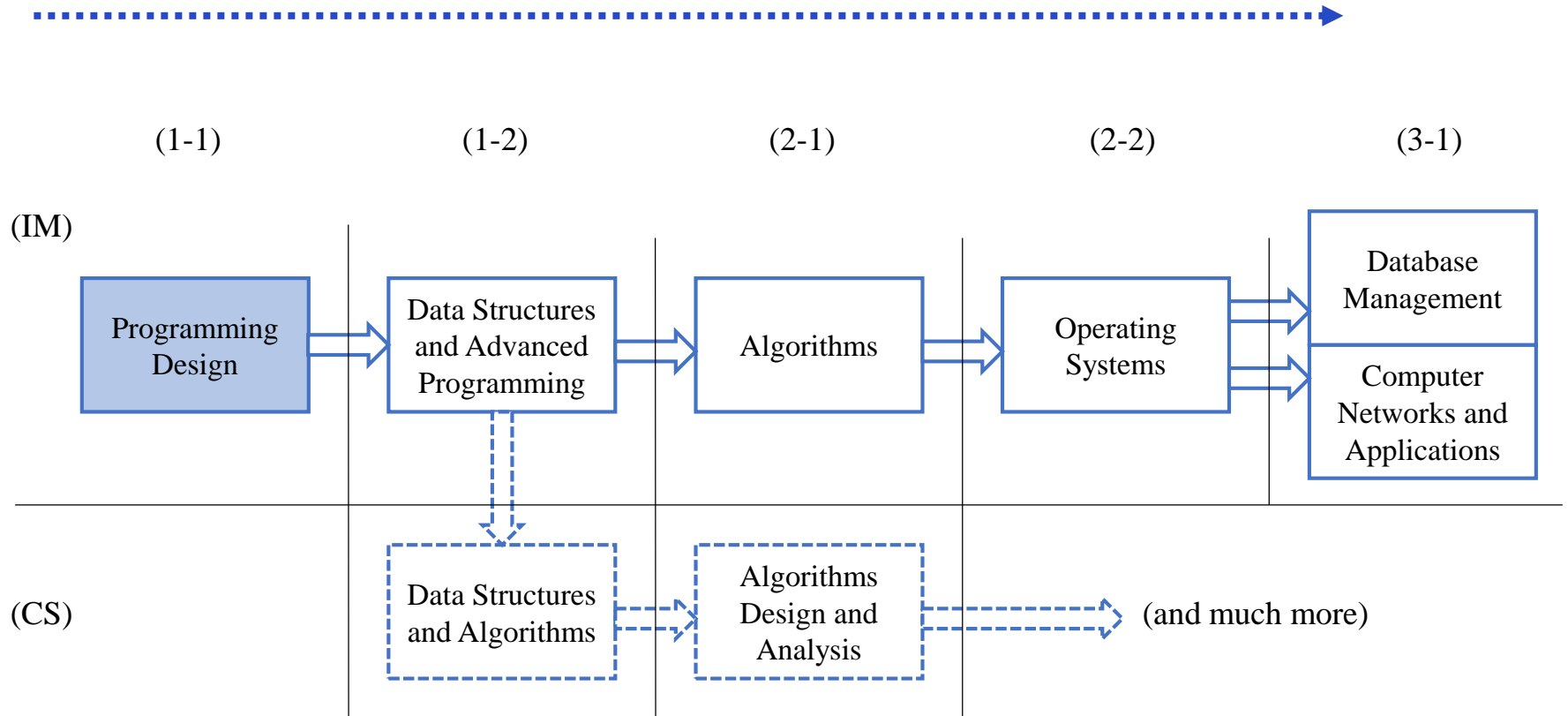
You really learned a lot!



http://d.rimg.com.tw/s1/a/8d/ed/21442506788333_602_m.jpg

But do not forget...

Related courses



Next steps



- Other NTU courses.
- OCW and MOOCs.
 - NTU OCW : <http://ocw.aca.ntu.edu.tw/ntu-ocw>
 - NTU MOOCs : <https://www.coursera.org/taiwan>
- Operations Research (on Coursera soon)
- Interns.
- Industry or research projects in school.
- Teaching others how to program.
 - [106-1](#) 、 [106-2](#) 、 [109-1](#)

Last words



- Practice makes perfect.
- Technologies change; programming principles do not.
 - At least do not easily change.
- Programming languages change.
 - Keep learning until you die (or retire).
- Make yourself be able to learn new things. And that means:
 - Study English.
 - Study mathematics.

你說得都很好，
可是聽起來又難又累...

Do it now



要種一顆樹，
最好的時間是二十年前，
或者是現在。

(網路名言)

Do it now, and keep going



***Most people overestimate what they can
do in one year and underestimate
what they can do in ten years.***

– Bill Gates (maybe)

Do it now, and keep going



Do it now, and keep going



最後的建議



- 當個獨特的人
 - 跨領域
 - 別人覺得你不用做的事
- 當個會表達的人
 - 簡報表達
 - 說出心裡的話
- 當個溫暖的人
 - 善待別人
 - 善待自己

*That's all.
Thank you!*



<https://www.facebook.com/helloiampeach>