

# 基礎程式設計

# Python

俞伯傑

R5524/Office Hour: Wed@15:00-16:00

[pcyu@g.yzu.edu.tw](mailto:pcyu@g.yzu.edu.tw)

1	課程簡介，資訊工程與程式語言簡介	Introduction of computer science and programming
2	Python編譯器與程式碼編輯器	Compiler and editor
3	基本語法	Syntax
4	變數與字串,運算符號	Variable types, operators
5	程式結構: 條件判斷式與迴圈(一)	Statement and loops (I)
6	程式結構: 條件判斷式與迴圈(二)	Statement and loops (II)
7	串列與字典 (I)	List and dictionary (I)
8	串列與字典 (II)	List and dictionary (II)
9	期中考週	Mid-term exam
10	日期與時間	Date and time
11	函數	Functions
12	模組與套件	Module
13	文件的輸入輸出	Files I/O
14	檔案處理	Files method
15	物件與類別 (I)	Class and objects (I)
16	物件與類別 (II)	Class and objects (II)
17	程式異常處理/(業界學長演講)	Exceptions/(Invited Talk)
18	期末考週	Final exam

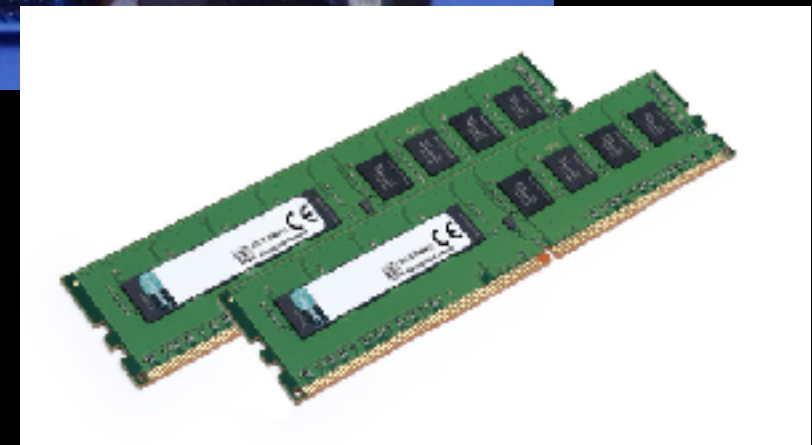
# 成績

- 期中考: 30%
- 期末考: 40%
- 出席/上課狀況/小考: 30%

# 電腦

- 硬體: input/output , 算術邏輯單元 , 記憶體
- 韌體: BIOS
- 軟體: 系統程式: 作業系統/編譯程式/組譯程式

應用程式: 遊戲/Office/繪圖軟體



# 單位

- Bit: 位元
- Byte: 位元組 = 8 bit (例如 'A'=01000001)
- KB: Kilo Byte = 1024 bytes
- MB: Mega Byte = 1024 KB
- GB: Giga Byte = 1024 MB
- TB: Tera Byte = 1024 GB

# 程式語言簡介

- 使用者用來與電腦溝通的語言
- 電腦只能理解0與1組成的形式
- 給予電腦指令，指令的集合 => 程式
- 透過不同的編譯/翻譯器，不同的程式語言都會被翻成相同的形式

# 程式語言簡介

- 低階語言:執行速度快，編寫困難  
ex: 機器語言，組合語言
- 高階語言: 語法較為簡單易懂  
ex: IDL/FORTRAN/BASIC/C/C++/JAVA/Python
- 資料庫查詢語言: SQL
- 人工智慧語言: 發展中 (LISP/Prolog)



# 程式語言簡介-機器語言

- 機器語言(machine language)是CPU可以直接解讀的指令集
- 都是由0與1組成的代碼
- 不同平台是不相通的
- 例如: 0000 (load), 0001 (store)
- 例如: 0000, 0000, 000000001000

# 程式語言簡介-組合語言

- 組合語言 (assembly language): 用英文簡寫來代表各種運算
- 例如: ADD, SUB
- 也是必須被翻譯成機器語言 (組譯器/assembler)
- ex:  $c = a + b$

mov ax, [a]

add ax, [b]

mov [c], ax

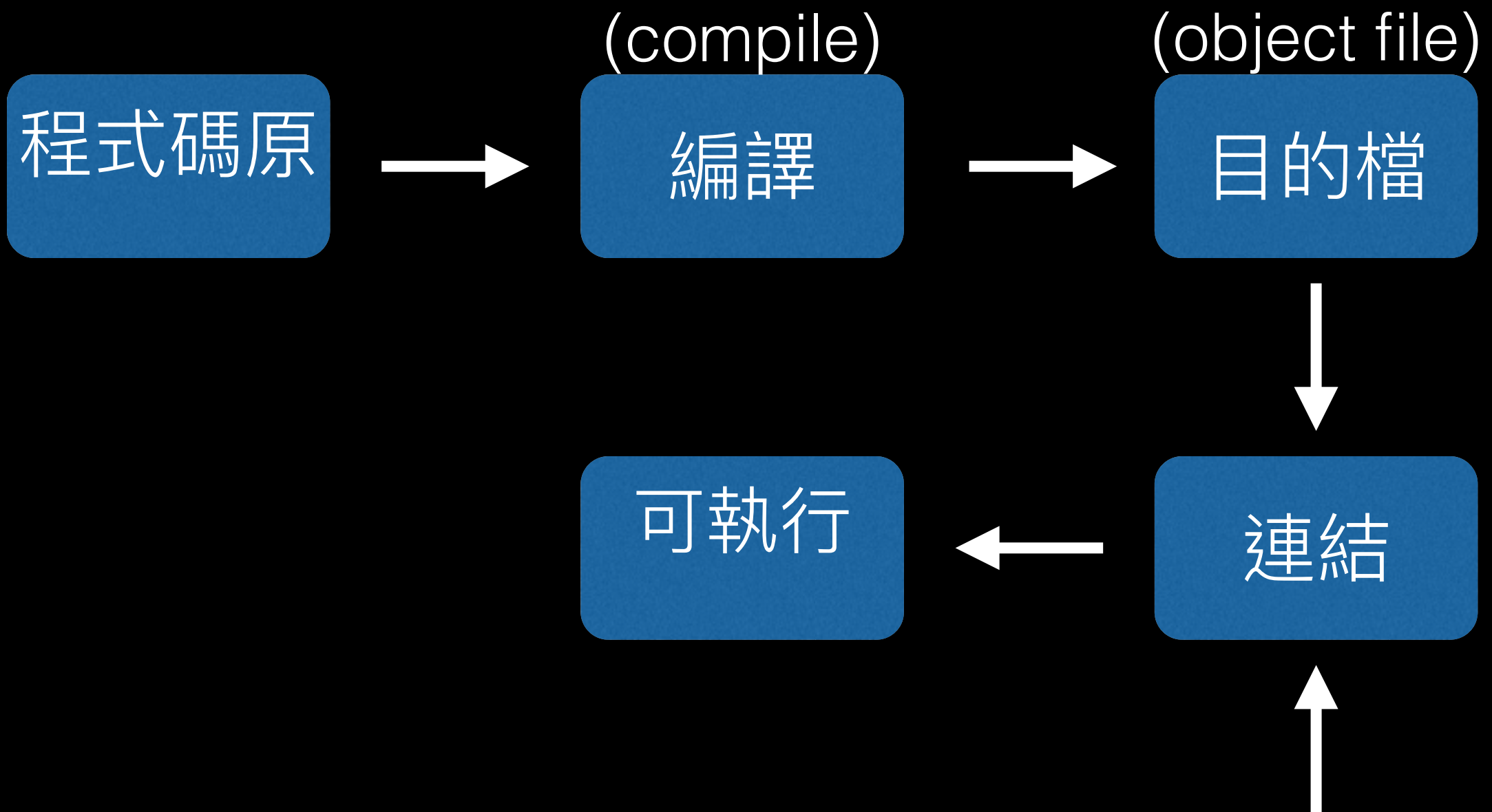
# 高階程式語言與應用

- FORTRAN/IDL: 處理數值計算，科學與工程應用
- BASIC與Visual Basic: 適合初學者的程式語言
- C/C++: 發展系統軟體或作業系統 (Unix)，物件導向功能 (C++)
- JAVA: 物件導向，資料處理，網際網路後相當熱門的語言，可以透過瀏覽器執行
- Python: 跨平台，容易擴充，Big Data

# 直譯 vs. 編譯

- 直譯式語言: 經過直譯器 (interpreter)，可以將原始程式的指令逐一翻譯執行，不需要經過編譯，如BASIC，HTML，Python
- 編譯式語言: 需要經過編譯器 (compiler)，輸出為執行檔，如C/C++

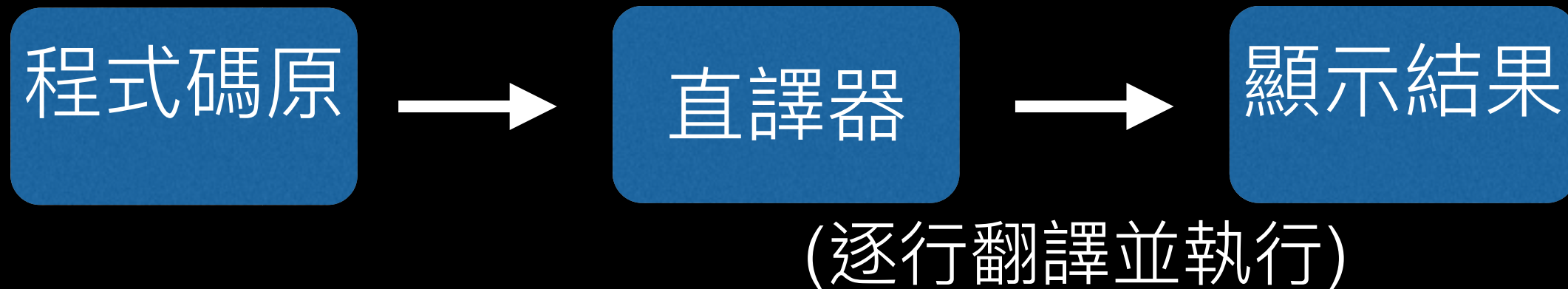
# 編譯式語言



編譯器: 例如gcc , g++ , VC++  
函式庫: 例如數學函式

其它目的檔  
或函式庫

# 直譯式語言



- 不產生目的檔或執行檔，會逐行翻譯並交付給電腦執行
- 每次都需要啟動直譯器
- 當程式碼某行發生錯誤，之前正確的部分仍然會被執行，而停留在錯誤的地方
- JavaScript，VBScript，Python

# FORTRAN

- Formula Translation
- 世界上最早的高階程式語言 (1954)
- 早期主要被用於從事科學計算
- 工程，數學與科學領域廣受歡迎
- 檔案處理能力不強，也沒有結構化的概念

# COBOL

- Common Business Oriented Language
- 通用商業語言，商用資料處理
- 1959年發表，以英文為主，用英文書寫陳述

```
//COBUCLG JOB CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1)
//HELLOWRLD EXEC COBUCLG,PARM.COB-'MAP,LIST,LET'
//COB.SYSIN DD *
001 IDENTIFICATION DIVISION.
002 PROGRAM-ID. 'HELLO'.
003 ENVIRONMENT DIVISION.
004 CONFIGURATION SECTION.
005 SOURCE-COMPUTER. IBM-360.
006 OBJECT-COMPUTER. IBM-360.
0065 SPECIAL-NAMES.
0066     CONSOLE IS CNSL.
007 DATA DIVISION.
008 WORKING STORAGE SECTION.
009 77 HELLO-CONST PIC X(12) VALUE 'HELLO, WORLD'.
075 PROCEDURE DIVISION.
090 000-DISPLAY.
100     DISPLAY HELLO-CONST UPON CNSL.
110     STOP RUN.
//LKED.SYSLIB DD DSN=SYS1.COBLIB,DISP=SHR
//          DD DSN=SYS1.LINKLIB,DISP=SHR
//CO.SYSPRINT DD SYSOUT=A
//
```



# BASIC

- Beginner's All-purpose Symbolic Instruction Code
- 適合初學者的語言
- 1960年代創立，簡單易學
- Visual Basic: 以BASIC為基礎發展，發展微軟視窗應用軟體的快速開發程式

# C/C++

- C語言: 1972年貝爾實驗室設計出C語言，並以此寫出早期UNIX作業系統
- 1983年加入類別與物件導向的概念，設計出C++
- 語法簡潔，執行效率高，跨平台

# Java

- 起源於1991年 (Oak)
- 動態網頁，使Web page能與使用者互動
- 物件導向的語言
- 但C語言的指標或是動態記憶體配置都沒有

# Python

- 早在1991年釋出，目前已是 Python 3.X
- 直譯式，物件導向
- 跨平台
- 簡潔，易讀，容易維護

- 安裝Python開發環境 - Anaconda
- Mac/Unix/Windows