WEEK04 – COMPUTER SOFTWARE

WEN-BIN JIAN

DEPARTMENT OF ELECTROPHYSICS, NATIONAL CHIAO TUNG UNIVERSITY

OUTLINE

1. Batch File & Programming
2. Program Structures
3. System/Application Programs
4. Interpreted/Compiled Language
5. Generations of Programming Language
6. Essential Application Programs

Exercise

CODING/PROGRAMMING – BATCH FILE PROGRAMMING

The MS-DOS Emulator in Windows 10

- Try to secure your file operations in a specified directory
- Start-up menu -> Windows System -> Command Prompt (right click mouse to open the folder where the file is placed in), it's a link file (.lnk), copy it and change directory to your working directory
- Edit a batch (.bat) file
- Call Command Prompt & run the batch file
- Input parameters to run a batch file: abc john 2 -> %1=john, %2=2
  - echo off - do not show message
  - operating the string – echo %para:~2,1%, echo %para:~-2%

IC_W401.bat

```
dir /ah
dir /s
mkdir aloha
dir *.* > "aloha/dirs.txt"
```

IC_W402.bat

```
@echo off
cls
echo %1 "%2"
set para=%1
echo %para%
echo %para:~2,2%
```

Ref1: https://www.computerhope.com/sethlp.htm

CODING/PROGRAMMING – BATCH FILE PROGRAMMING

The MS-DOS Emulator in Windows 10

- Declare Variables:
  - set var1=STRING, set var1=65535 (all contents are strings)
- Input/Output:
  - set /P var1=Input a number:, set /P var1=Input a string
  - echo Hello %var1%, (%var1% is the content stored in the variable 'var1')
  - pause, (waiting for the user's key pressed)
- Calculations:
  - set /A var1=var1+5, +, -, *, /, %, set /A var1+=5, set /A var1 %= 4 **(command lines)**
  - set /A var1=var1%%10, set /A var1="%var1% %% 10" **(batch file)**
  - set /A var1=(var1-10)*5+(var1-var2)*3 (grouping)

CODING/PROGRAMMING – BATCH FILE PROGRAMMING The MS-DOS Emulator in Windows 10

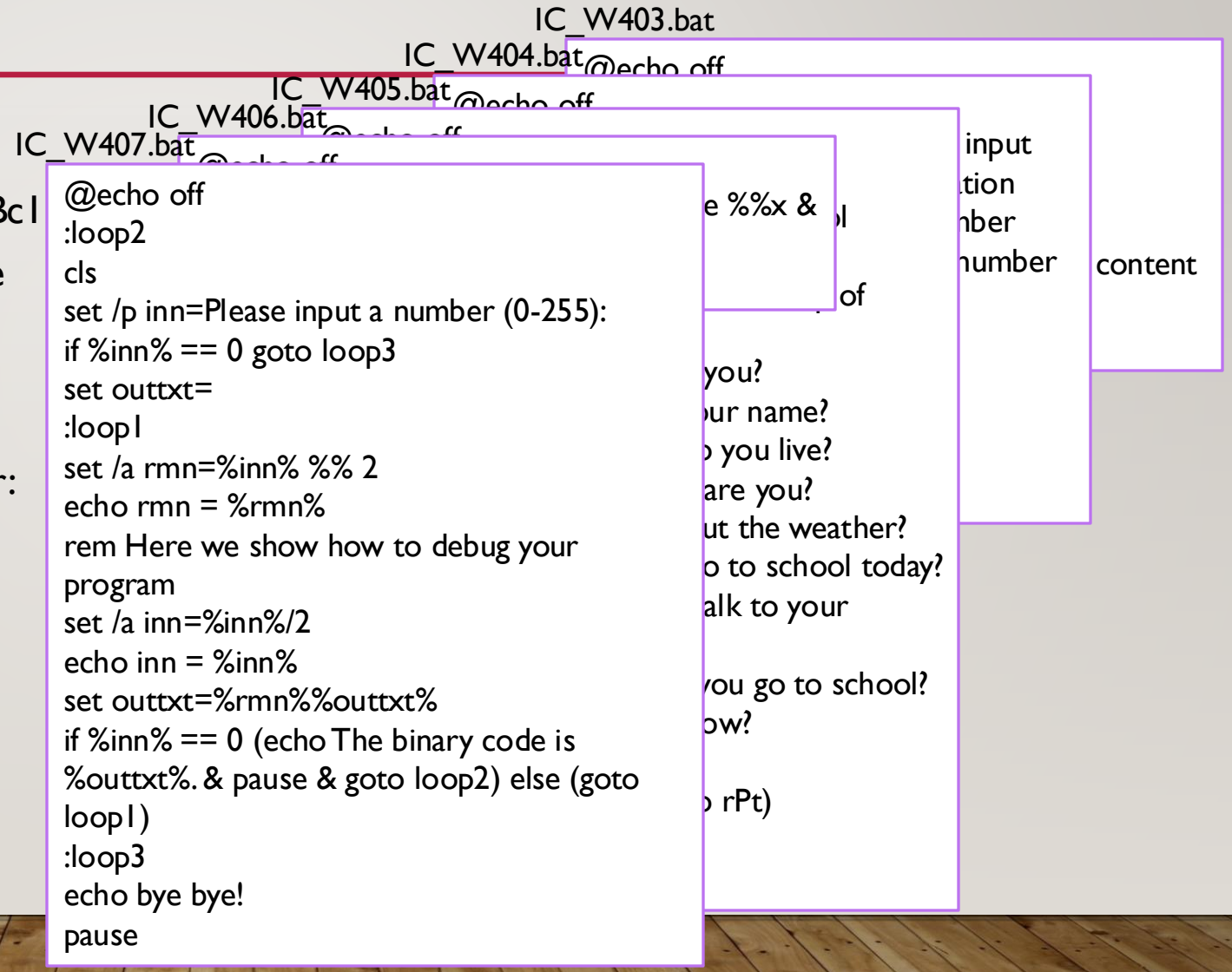
- bitwise operation: >>, <<, &(and), |(or), ^(xor)
- set /a var3 = "%var1% & %var2%", set /a var2 = "%var1% >> 2"

- String Operations:
  - %var1:ab=cd%, %var1:~n1, n2%, %var1:~n1,-n2%, %var1:~-n1%, %var1:~0,-1%, %var1:~1%
- Flow Control:
  - REM: remarks
  - if %var1% EQU %var2% (command & command & ...) else (command & command)
  - NOT, NEQ, LEQ, GEQ, LSS, GTR
  - Label: ":label_name", goto label_name
  - for %i in (0,1,9) do (command & command & ...) **(command lines)**
  - for %%i in (0,1,9) do (command & command & ...) **(batch file)**
  - ~~setlocal enabledelayedexpansion, for %%i in (0,1,9) do (set var1=!var1!+%%i)~~

Exercise

PROGRAM STRUCTURES – BATCH FILE PROGRAMMING

- Variables
  - Names: "aBc12", Use: content %aBc1
  - rem: comments / remarks in a line
- Output / Input
  - echo %var_name%
  - set /P var_name = Input a number:
  - pause
- Calculations & String Operations
- Others, e.g. redirection (>, >>), call
- Flow Control – if, for, goto



Ref1: https://www.computerhope.com/sethlp.htm

SYSTEM SOFTWARE / APPLICATION SOFTWARE

- System Software: provided by the operating system
  - process manager (multiplexing)
  - control panel, disk manager, event dealer
- Application Software: use computer to solve any other problems
  - filezilla ftp client, web page browser, email / server
  - MS Office: word, excel, powerpoint, access
  - paint.net, photoshop, autocad, sketchup, mathematica, latex
  - origin, SDK (software development kits), IDE (integrated development environment)

Ref1:

PROGRAMMING LANGUAGE

- **Interpreted Language** – real time command execution, run programs in another program, slow speed
  - single computer programming – access hardware easily / internet programming, need server to share hardware
  - Quick Basic – Get QBasic from Microsoft Store, it's a quick basic "interpreter" (focus on short, easy programs), Python (script, extensively used and developed)
  - Javascript – run by browser, php, MySQL database – easily run programs cross many platforms
- **Compiled Language** – source codes compiled to form an execution file of machine codes, the machine codes for UNIX, Windows, and macOS are all different
  - Microsoft Macro Assembler
  - C/C++, Microsoft Visual C++, Turbo C++, gcc, Xcode, Objective-C, Dev C++ – you can compile your programming codes in different OSs (UNIX, Windows, macOS) , limitation in cross platform program developments, run much faster than the interpreted and the java virtual machine codes
  - Java – define its own virtual machine codes, the virtual machine codes can be run across several different platforms – virtual machine codes run by programs on different OS platforms

Ref1:

PROGRAMMING LANGUAGE – PARADIGM

- **Imperative programming**: follow the design style of machine coding
- **Procedural programming**: take specified steps to reach a desired goal
- **Declarative programming**: emphasize program logic rather than control flow
- **Functional programming**: treat programs as evaluating mathematical functions
- **Object-oriented programming**: emphasize data structures with their own interface functions
- **Event-driven programming**: use event handling for control flow
- **Automata-based programming**: treat program as models of machines

Ref1: https://en.wikipedia.org/wiki/Comparison_of_programming_paradigms

PROGRAMMING LANGUAGE

- The 1st generation language (**1GL**) – machine language, the code is fast and efficient
- The 2nd generation language (**2GL**) – macro assembly language, language is specific to a processor
- The 3rd generation language (**3GL**) – high level language, programmer friendly, fortran, cobol, c, c++, object-oriented, java, basic, pascal
- The 4th generation language (**4GL**) – very high level language, python, ruby, and perl are between 3GL and 4GL
- The 5th generation language (**5GL**) – used in artificial intelligence research, OPS5, mercury

Ref1: https://en.wikipedia.org/wiki/First-generation_programming_language

PROGRAMMING **GOALS**

- Machine control: low level language, real time control, e.g. to process interrupt in nano seconds, use assembly, c++, java; labview is a high-level language used to control machine as well
- General purpose: c++, java
- Computer simulation: fortran, c++, java
- Data processing, statistical analysis: python, R
- Matrix operation: matlab
- Analytic Calculation: mathematica
- Business: cobol
- Coding of art, 3D & OpenGL: processing, p5.js (library: opencv)
- Internet: html5, javascript, php, sql language for web-based database
- Artificial intelligence: OPS5, mercury, lisp, prolog (library: tensorflow)
- Game: c++, java, html5, css3, javascript, SQL
- Android phone: java

Most of introduction courses for programming use c/c++ to write console programs.

Exercise

PROGRAMMING LANGUAGE – PYTHON SCRIPT

- Google search python3
- Download the latest version from https://www.python.org/downloads/, Download Windows x86-64 executable installer
- Install the program, check "add python 3.7 to path", open dos console and type python
- Install libraries in dos console: pip install numpy, pip install wheel, pip install matplotlib
- Start to write your programs

IC_W408.py

```
import os
import numpy as np
import matplotlib.pyplot as plt

route = 866
print(route, type(route))
cities = ["Tainan", "Taoyuan", "Taipei", "Taichung", "Hsinchu", "Chiayi"]
for city in cities:
        print(city, "is a wonderful city.")
os.system("pause")

x = np.random.rand(100)
y = np.random.rand(100)
plt.scatter(x,y)
plt.title("Scatter Plot")
plt.xlabel("X Value")
plt.ylabel("Y Value")
plt.show()
```

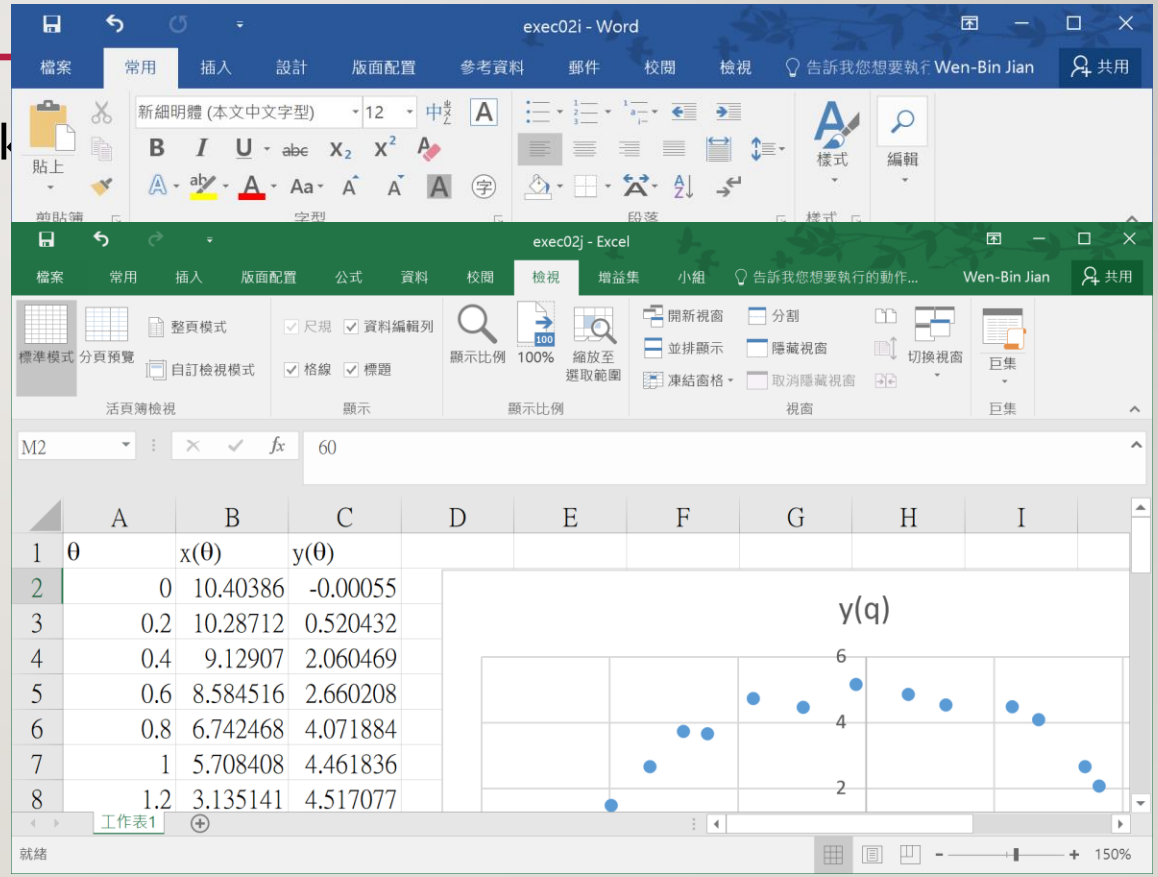
Ln: 3 Col: 4

Ref1:

Exercise

ESSENTIAL APPLICATION PROGRAMS

- Word, Excel, Powerpoint, Acess, Outlook
- Browser
- WinRar, Paint.Net(GIMP)
- SDK, IDE
- Mathematica, Matlab, Labview



Ref1:

DOS BATCH

WEN-BIN JIAN

DOS BATCH ENVIRONMENTS

- Working directory(工作目錄)?
- Cmd.exe 建立捷徑
  - Windows 工具
  - 按右鍵-開啟檔案位置(目錄)
  - 複製捷徑到工作目錄
  - 按右鍵-內容
  - 更改開始位置(目錄) 拷貝其他在該目錄中檔案的位置
- 啟動pspad撰寫程式 儲存檔案到工作目錄
  - 存檔時須選擇存檔類型為所有檔案，檔案名稱加上副檔名.bat

DOS BATCH COMMANDS

- @echo off 關掉與對話
- pause 暫停
- echo 此為dos batch的print功能
  - @echo off
  - echo Hello World!
  - pause
  - ------存檔後啟用命令提示字元，dir找看看有沒有.bat檔案，執行看看
  - 進入 命令提示字元 環境下比較容易debug看到錯誤

DOS BATCH COMMANDS

- 使用者輸入 輸入皆為字串
  - set /p va01=Your name:
  - =號後面為顯示給使用者看的提示文字
  - =號前面為變數名稱
  - 此後要看變數名稱va01的內容(字串)，都需要使用%va01%，變數內容前後都可以接上其他字串
  - set va02=123%va01% 321 設定另一個變數之字串內容
- 例子
  - @echo off
  - set /p va01="Please Input Your Name:"
  - echo How are you, %va01%?
  - set va02=123%va01% 321
  - echo va02=%va02%, %va02%%va01%12%va02%
  - pause

DOS BATCH COMMANDS-STRING OPERATION

- 取代字串 %va01:??=??%
  - %va01:1=2%把va01變數內字串中字元1換成2，%va01:ab=z%遇到ab兩個字換成z
- 子字串substring %va01:~??%
  - 字串字元位置編號為最左邊第0個字元，依序第1、第2、第3個字元
  - %va01:~n%取自第n個字元開始之後所有字元為字串
  - %va01:~n,m%取自第n個字元開始之後共m個字元為字串
  - %va01:~-n%取自倒數第n個字元開始之後所有字元為字串
  - %va01:~n,-m%取自第n個字元開始直到倒數第m的前一個字元結束中間的所有字元為字串
    如果沒有適當字串顯示ECHO 已關閉的錯誤訊息

DOS BATCH COMMANDS-IF判斷與迴圈

- :label 可以跳躍到的程式列位置
- goto label程式指令跳躍到label處繼續執行程式
- 以上兩個指令會造成無窮迴圈無法停止，因此需要加入if判斷指令
- if %va01% EQU LE (echo %va01% = LE) 可以比較變數存放的字串是否等於“LE”
- EQU等於、GTR大於、LSS小於
- ctrl+c可以結束無窮迴圈

```
@echo off
:loop1
set /p va01="Please Input A String:"
if %va01% EQU LE (echo %va01% EQU LE)
if %va01% GTR LE echo %va01% GTR LE
if %va01% LSS LE echo %va01% LSS LE
goto loop1
pause
```

DOS BATCH COMMANDS-STRING

- 請使用者輸入字串，譬如NYCU，一個字元一列印出來

```
set /p va01="Please Input A String:"
:loop1
echo %va01:~0,1%
set va01=%va01:~1%
if "%va01%" NEQ "" goto loop1
pause
```

- 請使用者輸入字串，譬如NYCU，倒過來印出字串UCYN

```
set /p va01="Please Input A String:"
set va02
:loop1
echo %va01:~-1%
set va02=%va02%%va01:~-1%
set va01=%va01:~0,-1%
if "%va01%" NEQ "" goto loop1
echo %va02%
pause
```

DOS BATCH COMMANDS-CALCULATION

- set /a va01=(10*20–25)/4
- set /a 啟動等號右邊的數值計算
- 加+, 減-, 乘*, 除/, 括號()
- 求餘數 %%

```
:loop1
set /p va01="Please Input A Number:"
set /a va02=va01*va01
echo %va01%*%va01%=%va02%
set /a va03=2*%va01%*%va01%-3*%va01%+4
set /a va04="2*%va01%*%va01%-3*%va01%+4"
set /a va05=(10 * 20 - 25) / 4
echo %va03%, %va04%, %va05%
goto loop1
```

```
:loop1
set /p va01="Please Input A Number:"
set /a va02=va01%%8
set /a va03=va01/8
echo %va01% mod 8 = %va02%, %va01% / 8 = %va03%
goto loop1
```

DOS BATCH COMMANDS-CALCULATION

- 利用除法與求餘數計算數字的2, 8進位表示法
- 讀取一個數字
- 取得數字的2或8餘數，將此數字結合到輸出字串左邊
- 把此數字對2或8做整數除法
- 運算後如果此數字還大於零就繼續上面求餘數與除法計算

```
:loop2
set /p va01="Please input a number:"
set output=
:loop1
set /a rmn=va01%%2
set output=%rmn%%output%
set /a va01=va01/2
if %va01% gtr 0 goto loop1
echo %output%
goto loop2
```

DOS BATCH COMMANDS-CALCULATION

- 輸入2或8進位數字，轉換成10進位數字
  - 使用者輸入2或8進位數字
  - 字串處理，最右邊為最低位數
  - 從右到左讀取位數
  - 準備2^n，或是8^n的數乘上該位數
  - 加總數字

```
:loop2
set /p va01="Please input an octal number:"
set sum=0
set exp=1
:loop1
set dig=%va01:~-1%
set va01=%va01:~0,-1%
set /a sum=sum+dig*exp
set /a exp=exp*8
if "%va01%" NEQ "" goto loop1
echo %sum%
goto loop2
```

DOS BATCH COMMANDS-CALCULATION

- N!，N階乘計算，或是1+2+3+…+N的計算
  - 使用者輸入數字N
  - 用變數一開始是1乘上數字N
  - 把數字N減1，然後重複上一列運算
- 陣列式畫星星
  - 輸入數字N，起始一個*字串
  - 增加一個*並把數字N減1
- 巢狀式雙迴圈用來解決
  - 九九乘法表

```
:loop2
set /p va01="Please input a number:"
set num=%va01%
set tot=1
:loop1
set /a tot=tot*num
set /a num=num-1
if %num% gtr 0 goto loop1
echo %va01%!=%tot%
goto loop2
```

只能算到16!，17!變負數
超過32bit有號整數範圍

```
:loop2
set /p va01="Please input a number:"
set istr=*
:loop1
echo %istr%
set istr=%istr%*
set /a va01=va01-1
if %va01% GTR 0 goto loop1
goto loop2
```

DOS BATCH COMMANDS-CALC&STRING

- 結合數字運算與字串處理及雙迴圈
- 使用者輸入字串
- 第一個字元在第一列印一次
- 第二個字元在第二列印二次
  - 要另外用一個數字計入現在是第n個字元
  - 使用迴圈把第n個字元擴增成n個相同字元字串

```
set /p var1=Please Input a String:
set itr=1
:loop2
set chr=%var1:~0,1%
set var1=%var1:~1%
set cntr=0
set out=
:loop1
set out=%out%%chr%
set /a cntr=cntr+1
if %cntr% lss %itr% goto loop1
echo %out%
set /a itr=itr+1
if "%var1%" neq "" goto loop2
pause
```

EXERCISE

1. Please use the DOS batch file for programming. Please write a program to get two decimal numbers from users and transform them into binary numbers. Then, calculate the “and” operation between the two numbers and show the result in binary form.
2. Please use the DOS batch file for programming. Please ask the user to give you an octal number and convert the octal number to decimal, hexadecimal, and binary numbers.
3. Please use the DOS batch file for programming. Please ask the user to input a string and reply to the user about how many characters of ‘a’ exist in the string.

EXERCISE

```
cls
set /p inn=Please input a number (0-255):
set /a rmn=%inn%%%2
echo rmn = %rmn%
set /a inn=%inn%/2
if %inn% == 0 (echo We finish the calculation.)
```

1. Please explain what it has done in the codes of a DOS batch file shown in the right.
2. Please use the DOS batch file for programming. Please ask the user to input a number N and print out one stars in the first row, two stars in the 2nd row, …, and N stars in the Nth row.
3. Please use the DOS batch file for programming. Please ask the user to input a string and print out the even and odd number of the string characters.

EXERCISE

1. Please use the DOS batch file for programming. Please find the reverse of the user's name and print out 'I found that your name in reverse is eman_resu.'
2. Please use the DOS batch file for programming. Please ask a number N from the user and reply the result of N!=N*(N-1)*…*1.

EXERCISE

1. Please use MS Word to prepare the best appearance of your curriculum vitae.
2. Please use MS Excel to draw scattering plot of the xy data: (14.2, 21.5), (16.4, 32.5), (11.9, 18.5), (15.2, 33.2), (18.5, 40.6), (22.1, 52.2), (19.4, 41.2), (25.1, 61.4). Please draw a line of least square fitting to the data.