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

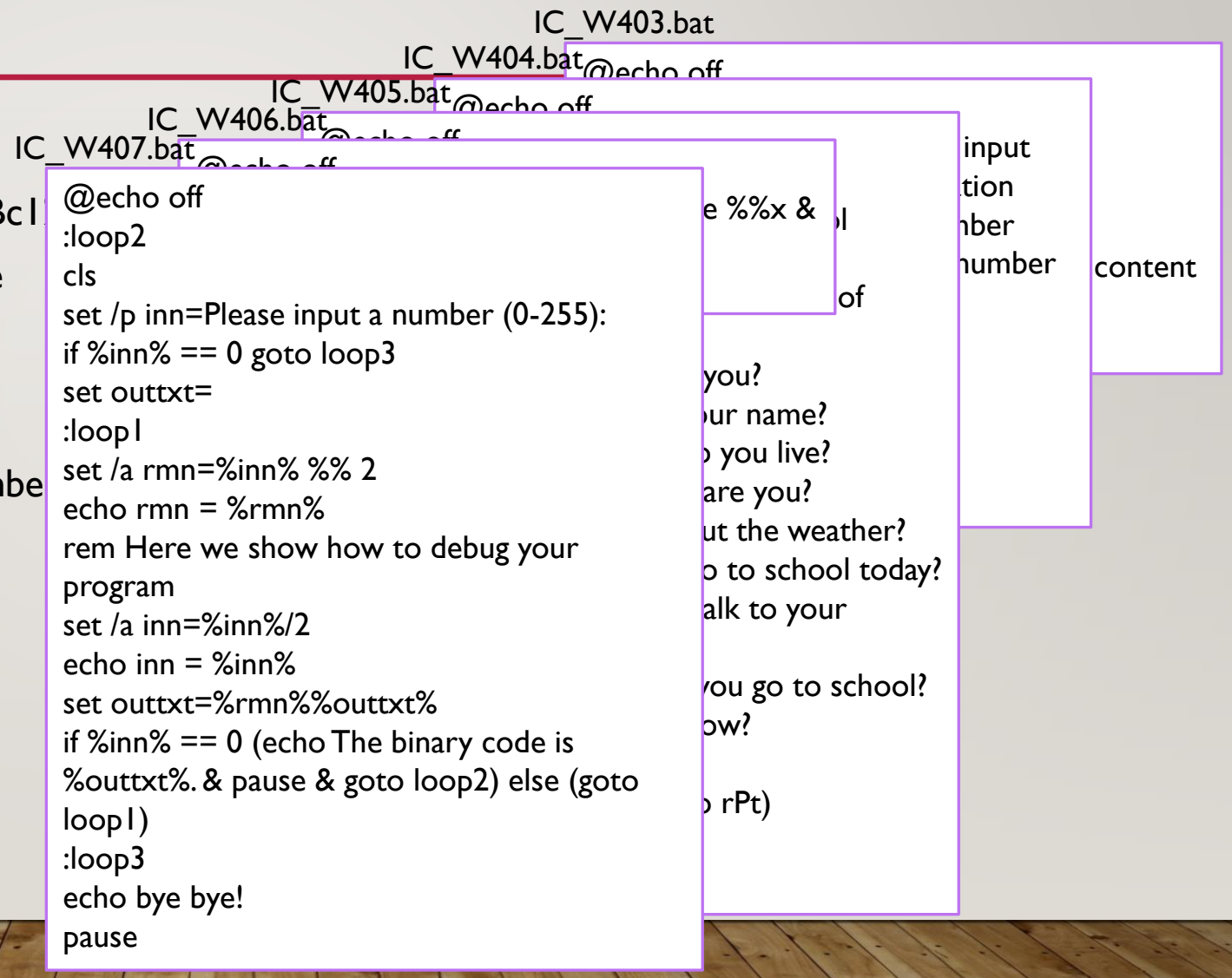
- bitwise operation: >>, <<, &(and), |(or), ^(xor)
- set /a var3 = "%var1% & %var2%", set /a var2 = "%var1% >> 2"

- 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 numbe
  - 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

Ref1:

IC_W408.py

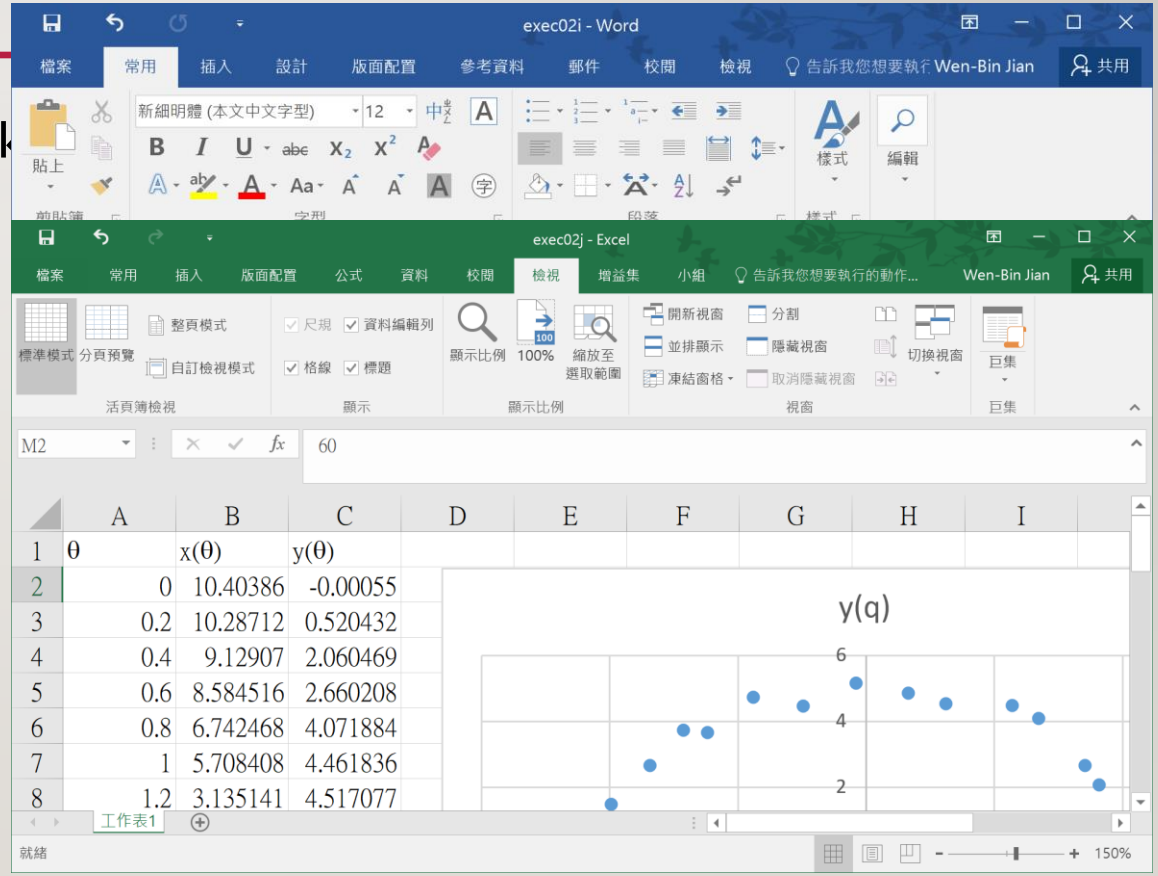
```
import os
import numpy as np
import matplotlib.pyplot as plt

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

Exercise

ESSENTIAL APPLICATION PROGRAMS

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



Ref1:

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.