計算機系統與應用
— How Computer Programming Work
— Introduction to C & Flow Chart

陳秋男

cnchen@pu.edu.tw
http://www1.pu.edu.tw/~cnchen/cs2003/
How Computer Running

- Hardware
  - CPU
  - Chip sets
  - Firmware

- Software
  - Operation System
  - System Software
  - Applications

- I/O Devices
Where Software Is

- Cellular Phone
- Motors
- Train Station
- Airplane
- Traffic Light
- Home
- ...

[Image of a woman talking on a phone and a busy street scene]
What Is Computer’s Language

- Binary Code
- Machine Code
- Assembly Code
- High Level Programming Language
- Scripts
- Markup Language
- Standard Query Language
Programming Steps

1. Source Code
   - Editor

2. Object Code
   - Macro Pre-processor
   - Assembler
   - Complier
   - Interpreter

3. Executable Code
   - Linker
   - Loader
C Programming Process

- Start
- Editing source program
  - Compiler
    - Object module
      - Linker
        - Executing
          - Library
            - Object module
              - Error

- Error
- Done
Editing Source Program

- `pico hello.c`

```c
#include <stdio.h>

void main(void)
{
    printf("Hello, world!\n");
}
```
Compiler

- `gcc hello.c`
  - `a.out`
- `gcc -o hello hello.c`
  - `hello`
  - `gcc -o object-filename source-filename`
- Warning.
- Error.
Structured Programming

- One-Enter-Point / One-Exit-Point Rule
- Without Unconditional GOTO
- Fundamental Control Structures
  - Sequence Control Structure
  - Selection Control Structure
  - Looping Control Structure
    - Do While
    - Do Until
Sequence Control Structure

Enter

Procedure 1

Procedure 2

Procedure 3

Exit
Selection Control Structure

Enter

True
Decision ?

Procedure 1

False
Procedure 2

Exit
Looping Control Structure

Do While
Enter
Decision ?
True
Procedure
False
Exit

Do Until
Enter
Decision ?
True
Procedure
False
Exit
Flow Chart

- Start / End of Program
- Input / Output of Data
- Data Processing / Procedure
- Sub-program
Flow Chart

Decision

Connect Point

Program Flow

Reference
Flow Chart

- Input one student’s homework scores and total them.
Pseudo-code

- // Program: Sum and Average
- // Author: C.N. Chen
- // Course: C3

void main ()
{
    Input hw1, hw2;
    // sum of hw1 and hw2
    sum = hw1 + hw2;
    // average of total score
    average = sum / 2;
    if (average < 60)
    {
        Output pass;
    }
    else
    
    Output failure;
}
SFC - A Structured Flow Chart Editor

http://www.cs.sonoma.edu/~tiawatts/SFC/
An Example of C

/* My 2nd C Program */
#include <stdio.h>
void main(void)
{
    int i;
    i = 2;
    printf("Hi, C!\n");
    printf("This is my %dnd C program! …", i);
    printf("OK.");
}
/* */ and #

- Explanatory Notes
  - /* My 2nd C Program */
  - C++: //

- Preprocess – cpp
  - #include
    - <stdio.h>
    - <stdlib.h>
  - #define
main ( )

- Main Program
  - A C program is a collection of functions.
  - Begin form `main( )` function.
  - `void main(void)`
  - An example of `main()`
    - `void main(void)`
    - {
      - Statements….
      - ...........
    - }
{} and ;

- Statements
  - End with ;
  - {} – a group of statements.

- Usual Statements
  - Variable Declarations & Value Assignments
  - Algorithmic Operations
  - Function calls
  - Flow Control
  - And so on…

```c
{ int i;
  i = 2;
  printf("Hi, C\n");
  printf("This is my %dnd C program! ...", i);
  printf("OK.");
}
```
Variable Declarations & Value Assignments

- `int i ;`
  - `i` - variable name
  - `int` - data type

- `i = 2;`
  - `2` - value
  - `=` - assignment
Standard Input/Output

- C has no ability to do I/O operation but through *I/O Function Library*.
- Hardware dependent
  - Ex: `printf()` will not be processed until linking.
  - Portability
- Redirection
  - `Prog > file`
  - `Prog >> file`
  - `Prog < file`
stdio.h

- **printf()**
  - Output formatted data to the standard I/O.

- **scanf()**
  - Input formatted data from the standard I/O.

- **getchar()**
  - Read one character from the standard I/O.

- **putchar()**
  - Write one character to the standard I/O.

- **EOF**
  - -1
printf () Function

- printf() is a external function and store in the Function Library.
  - #include <stdio.h>
- \n  - print out a new line.
- %d  - print out the value of variable.
  - %  - formatted
  - "Hi, C!\n" 

Print out a message on the stand-output (terminal screen).
printf () Practice

- `#include <stdio.h>`
- `main()`
- `{`
- `printf("*\n");`
- `printf("**\n");`
- `printf("***\n");`
- `printf("****\n");`
- `printf("*****\n");`
- `printf("******\n");`
- `}`
printf = Print + Formatted

- `printf(“This is my %dnd C program! …”, i);`
- `printf(“…%d….%d…”, i, j);`
- ` %[f][w][p][i][x] ` formatted
  - length: data type of variable
  - precision: decimal length
  - width: total length
  - flag: signed & alignment

i = 2;
printf = Print + Formatted cont.

- **w.p** – column control
  - %8.2f
    - __ 1 2 3 . 4 5
  - %7.8f
    - illegal – out of control.

- **+** flag – signed
  - %+d
    - +123
    - -123

- **-** flag – left alignment
  - %5d
    - __ 1 2 3
  - %-5d
    - 1 2 3 __
<table>
<thead>
<tr>
<th>%c</th>
<th>字元</th>
</tr>
</thead>
<tbody>
<tr>
<td>%s</td>
<td>字串</td>
</tr>
<tr>
<td>%d</td>
<td>有號十進位整數</td>
</tr>
<tr>
<td>%e</td>
<td>浮點 (指數表示)</td>
</tr>
<tr>
<td>%f</td>
<td>浮點 (小數點表示)</td>
</tr>
</tbody>
</table>
#include <stdio.h>
main()
{
    float p = 1234.56;
    float n = -456.78;

    printf("Value of p is:%10.2f\n", p);
    printf("Value of p is:%10.5f\n", p);
    printf("Value of p is:%10.1f\n", p);
    printf("Value of p is:%10.7f\n", p);
    printf("Value of n is:%-10.2f\n", n);
    printf("Value of p is:%+10.2f\n", p);
    printf("Value of n is:%-10.2f\n", n);
    printf("Value of p is:%+10.2f\n", p);
    printf("Value of n is:%+10.2f\n", n);
    printf("Value of p is:%+-10.2f\n", p);
    printf("Value of n is:%+-10.2f\n", n);
}
**scanf () Function**

- `scanf("%c %d %f", &a1, &a2, &a3,);`
  - `%c`: Input a character to the variable.
  - `%d`: Input a integer number to the variable.
  - `%f`: Input a real number to the variable.

**Separator**

- `scanf("%c,%d,%f", &a1, &a2, &a3,);`
- `scanf("%c;%d;%f", &a1, &a2, &a3,);`
scanf () Practice

- #include <stdio.h>
- main()
- {
-   int i, j, k, l;
-   printf("Input 2 numbers: ");
-   scanf("%d %d", &i, &j);
-   printf("The product is %d\n", i * j);
-   printf("\nInput 2 numbers again (ex. 12;34): ");
-   scanf("%d;%d", &k, &l);
-   printf("The sum is %d\n", k + l);
- }