

CP 111 Computer Programming (2/2554 Section B01-B02)

Handout 1^{*}

หัวข้อ

- Introduction to C (Chapter 11)
- Variables in C (Chapter 12.2)
- Operators in C (Chapter 12.3)

Introduction to C

1. ภาษา (โปรแกรม) C = a general purpose “High-level” computer programming language
 - ไม่ขึ้นกับ ISA (Instruction Set Architecture) เช่น ภาษา C สำหรับ LC-3, IA-32 (Intel), ARM, etc มีรูปแบบการเขียนโปรแกรมเหมือนกัน
 - โปรแกรมที่เขียนขึ้นดูเข้าใจง่ายกว่าโปรแกรมภาษาแอสเซมบลี
 - ใช้ตัวแปร (Variables) สำหรับเก็บค่า (ตัวแปรสามารถตั้งชื่อได้ตามความเหมาะสม) แทนที่จะอ้างถึง Memory address หรือ Register อย่างในโปรแกรมภาษาแอสเซมบลี
 - ใช้โอเปอเรเตอร์หรือตัวกระทำ (Operator) สำหรับการสั่งให้ประมวลผลกับตัวแปร เช่น operator + - * /
2. ภาษา C เป็นภาษาโปรแกรมคอมพิวเตอร์แบบ Procedural (Procedural programming language)
 - โปรแกรมภาษา C หนึ่งๆ จะประกอบไปด้วยหลายๆ โพรซีเยอร์ (หรือ Subroutine)
 - ภาษา C จะเรียก โพรซีเยอร์ (หรือ Subroutine) ว่า Function
3. ภาษา C ถูกคิดค้นโดย Dennis Ritchie ที่ Bell lab, USA สำหรับใช้กับระบบปฏิบัติการ Unix (Linux)
 - มาตรฐานภาษา C ถูกกำหนดเมื่อ ปี 1988 เรียกว่า ANSI C (ANSI = American National Standards Institute)
 - ปี 1990: ปรับปรุงมาตรฐานเล็กน้อยโดย ISO เรียกมาตรฐาน C90 และอีกครั้งปี 2000 เรียกมาตรฐาน C99
 - ในวิชาจะเรียน ANSI C (และ C++ บางส่วน) ไม่ได้เจาะจงเน้นว่าเป็นภาษา C ที่ขึ้นอยู่กับ Tools อย่างเช่น Visual C++ หรือ Objective C (MAC OS X, iOS)
 - ดู Appendix D Textbook หรือ หนังสือของ Kernighan & Ritchie สำหรับแหล่งอ้างอิง
4. โครงสร้างโปรแกรมภาษา C เบื้องต้น

```
/* This is an example of C program.
   CP111
*/
#include <stdio.h>
#define PI 3.1416
int main()
{
    float a;
    a=pi/2;
    printf("a=%f\n", a);
    return 0;
}
```

^{*} เรียงเรียงจาก Slide วิชา ECE190 UIUC by V.Kindratenko (<https://wiki.engr.illinois.edu/display/ece190/Lectures>)

(4.1) Comments: ข้อความที่อยู่ภายในเครื่องหมาย

(4.2) Pre-processing directives เริ่มต้นด้วยเครื่องหมาย

- #include <stdio.h> หมายถึง
- #define PI 3.1416 หมายถึง
- อื่นๆ จะได้เรียนต่อไป #ifdef, #endif, #define สำหรับนิยาม Macro

(4.3) Recall: ภาษา C เป็น Procedural language โปรแกรมในหน้าที่แล้ว ประกอบด้วยฟังก์ชัน 1 ฟังก์ชันคือ main()

- ทุกๆโปรแกรมภาษา C แบบ ANSI จะต้องมีฟังก์ชัน main
- โปรแกรมจะเริ่มต้นทำงานที่ statement แรกของภาษา C
- รูปแบบการนิยามฟังก์ชัน (จะเรียนรายละเอียดอีก) คือชื่อฟังก์ชัน และตามด้วย body ในเครื่องหมาย { } เช่น

```
int main()
{
    ...
}
```

(4.4) Variables (ตัวแปร)

- ตัวแปรทุกตัวที่จะใช้ในฟังก์ชัน จะต้องถูกประกาศ (variable declaration) ที่ตอนต้นของฟังก์ชัน (ภาษา C++ อนุญาตให้อยู่ที่ใดก็ได้)
- ในโปรแกรมหน้าที่แล้วมีการประกาศและใช้ 1 ตัวแปร คือ
- การอ้างถึงตัวแปร (อ่านหรือเขียนค่า) ทำได้โดยอ้างชื่อ และมีข้อกำหนดในการตั้งชื่อตัวแปร

(4.5) ภาษา C มีชุดของ standard library function เช่น จัดการกับ input/output ในโปรแกรมข้างบนมีการใช้งานฟังก์ชัน printf สำหรับแสดงค่าของตัวแปร a ออกสู่หน้าจอ

(4.6) โปรแกรม Compiler จะทำหน้าที่ในการแปลงโปรแกรมภาษา C ให้กลายเป็นภาษาเครื่อง

C	Machine language (displayed as LC-3 ISA assembly language)
<pre>int main() { int A, B, C; A = 5; B = 10; C=A+B; ... }</pre>	<pre>AND R0, R0, #0 ADD R0, R0, #5 ST R0, A ; A = 5 AND R0, R0, #0 ADD R0, R0, #10 ST R0, B ; B = 10 LD R0, A LD R1, B ADD R0, R0, R1 ST R0, C ; C = A + B .END A .BLKW #1 ; int A B .BLKW #1 ; int B C .BLKW #1 ; int C</pre>

- ในวิชานี้ Target hardware คือ PC ซึ่งใช้ CPU ของ Intel ซึ่งมี ISA แบบ IA-32 (Intel)
- Compiler ที่จะแนะนำให้ใช้ในวิชานี้ คือ Visual C++ (Visual Studio) เวอร์ชัน 6.0 หรือ 2008
- C program compilation flow ดู Slide 11-5 to 11-6

Variables in C

5. Recall (ยังจำได้ไหม): ในโปรแกรมภาษาแอสเซมบลีเราจะใช้ memory สำหรับเก็บค่าที่เราจะประมวลผล และการอ้างถึงทำได้โดยการอ้างถึง Memory address (แต่เรามักจะใช้ directive ช่วย เช่น LD R0, A)

- ในภาษา C เราจะใช้ variable หรือ ตัวแปร ในการทำหน้าที่เหล่านี้
- ก่อนการใช้ตัวแปร ต้องมีการประกาศตัวแปร (มิฉะนั้นจะ compile โปรแกรมไม่ได้)

(5.1) รูปแบบ (syntax) การประกาศตัวแปร:

```
<type> <name>;  
เช่น      int val;  
หรือ      float theta;  
หรือ      double dist;  
หรือ      char in_ch;
```

(5.2) Three basic data types

int : 2's complement integer (โดยทั่วไปจะเป็น 32-bit integer)

สามารถกำหนด Type qualifiers: signed, unsigned, short, long เช่น

```
unsigned int a; /* a is in the range */  
signed int b; /*equiv. to int b */  
long int c;  
short int d;
```

float: 32-bit floating point

double: 64-bit floating point

char: 8-bit data นิยมใช้สำหรับเก็บรหัสแอสกีของตัวอักษร

(5.3) การตั้งชื่อตัวแปร

- ประกอบด้วย ตัวอักษรภาษาอังกฤษ ตัวเลข เครื่องหมาย _ (underscore)
- ตัวแรกห้ามเป็น ตัวเลข เช่น int 1234a; ไม่ได้ ต้องเป็น int _1234a;
- ห้ามตั้งชื่อซ้ำกับ reserved word (ดู appendix D.2.6) เช่น int if; ไม่ได้
- สามารถกำหนดค่าเริ่มต้น เช่น

```
int area=2;  
double pi=3.1416;  
char in_ch='a';
```

(5.4) ขอบเขตของตัวแปร: global หรือ local: ดู Slide 12-7 to 12-8 (จะสอนรายละเอียดภายหลัง)

(5.5) คำศัพท์ Literals = Constant values เช่น 87, 0x57, 1.345, 454.12e-10, 40.0e12, 'c'

Operators, Expressions, Statements in C

6. Operators ใช้สำหรับระบุการประมวลผล (manipulate) ค่าของตัวแปร

(6.1) Arithmetic Operators ได้แก่ +, -, *, /, %

(6.2) Assignment Operators ได้แก่ = หมายถึงการกำหนดค่าให้ตัวแปร (อื่นๆ += -= *=, เป็นต้น)

(6.3) Relational Operators ได้แก่ > (มากกว่า), >= (มากกว่าหรือเท่ากับ), <, <=, != (ไม่เท่ากับ), == (เท่ากับ)

(6.4) Logical Operators ได้แก่ || (OR), && (AND), ! (NOT)

(6.5) อื่นๆ จะสอนต่อไป

7. Expressions

- ข้อความที่ประกอบไปด้วย variables และ literals และ operators เช่น

x*y	x>2	x+2	(x>=2) && (x<=4)
-----	-----	-----	------------------

- สามารถคำนวณค่า (evaluate) ออกมาได้ เช่น

x*y จะมีค่าเท่ากับค่าปัจจุบันของตัวแปร x คูณกับค่าปัจจุบันของตัวแปร y

x>2 ผลลัพธ์ของการตรวจสอบ x ว่ามีค่ามากกว่า 2 หรือไม่ จะได้ค่า True (1) หรือ False (0)

(x>=2) && (x<=4) ผลลัพธ์ของการตรวจสอบ x ว่ามีค่ามากกว่าหรือเท่ากับ 2 และน้อยกว่าหรือเท่ากับ 4 หรือไม่ จะได้ค่า True (1) หรือ False (0)

8. Statements

- ข้อความที่ประกอบด้วย variables และ literals และ operators เพื่อแสดงสิ่งที่สั่งให้คอมพิวเตอร์ทำ

- 1 unit of work ต่อ 1 statement

- จบท้ายด้วย ; (semicolon)

- ตัวอย่าง

```
{
    int a,b,c,d;
    a = (b*b) + (d*d);
    /* Statement ข้างบนเป็นการสั่งให้กำหนดให้ตัวแปร a มีค่าเท่ากับ ..... */

    c = (b/d) + (a*d);
    /* Statement ข้างบนเป็นการสั่งให้กำหนดให้ตัวแปร c มีค่าเท่ากับ ..... */

    b=10;
    c = (b>=5);
    /* b>=d เป็น ..... ถ้า b มีค่าเท่ากับ 10 แล้ว ..... */
}
```

9. Basic I/O with printf และ scanf (จำการใช้งานไปก่อน)

(9.1) Output: แสดงผลลัพธ์ออกสู่หน้าจอโดยใช้ standard library function printf()

- รูปแบบ: printf(format_string);

- ตัวอย่าง

```
{
    int a;
    float b;
    double c;
    printf("Hello world\n"); /* '\n' = newline character literal */
    a=10;
    printf("%d\n",a); /* display the value of a onto the screen */
    b=12.34;
    printf("%f\n",b); /* display the value of b onto the screen */
    c=b+12.3;
    printf("a=%d b=%f c=%f \n",a,b,c);
}
```

(9.2) Input: รับค่าจากคีย์บอร์ดโดยใช้ standard library function scanf()

- รูปแบบ: scanf(format_string,&variable_name);

ตัวอย่าง

```
{
int aa;
float bb;
double cd;

    printf("Enter an integer:");
    scanf("%d",&aa);

    printf("Enter a float:");
    scanf("%f",&bb);

    printf("Enter a double:");
    scanf("%lf",&cc);

    printf("Enter an integer, a float and a double:");
    scanf("%d %f %lf",&aa,&bb,&cc);

    ...
}
```

(9.3) String format (บางส่วน)

%d หรือ %i - decimal (integer)	%x - Hexadecimal	%c – ASCII character
%f – floating point	%lf – long floating point (double)	%s – ASCII character string

10. Exercise

จงทำโปรแกรมภาษา C ข้างล่างให้สมบูรณ์ เพื่อสั่งให้คอมพิวเตอร์ทำการรับค่าแบบ integer (จำนวนเต็ม) จำนวน 2 ข้อมูล
หลังจากนำข้อมูลทั้งสองมาคูณกันแล้วแสดงผลพร้อมออกสู่หน้าจอ

```
#include <stdio.h>

int main()
{
int a,b,c;
    printf("Enter 1st value: ");
    scanf("    ",&a);
    printf(                );
    scanf(                );
    c=
    printf("%d x %d = %d\n",    ,    ,c);
    return 0;
```

