



บทที่ 3 โครงสร้างภาษาซีเบื้องต้น Part1

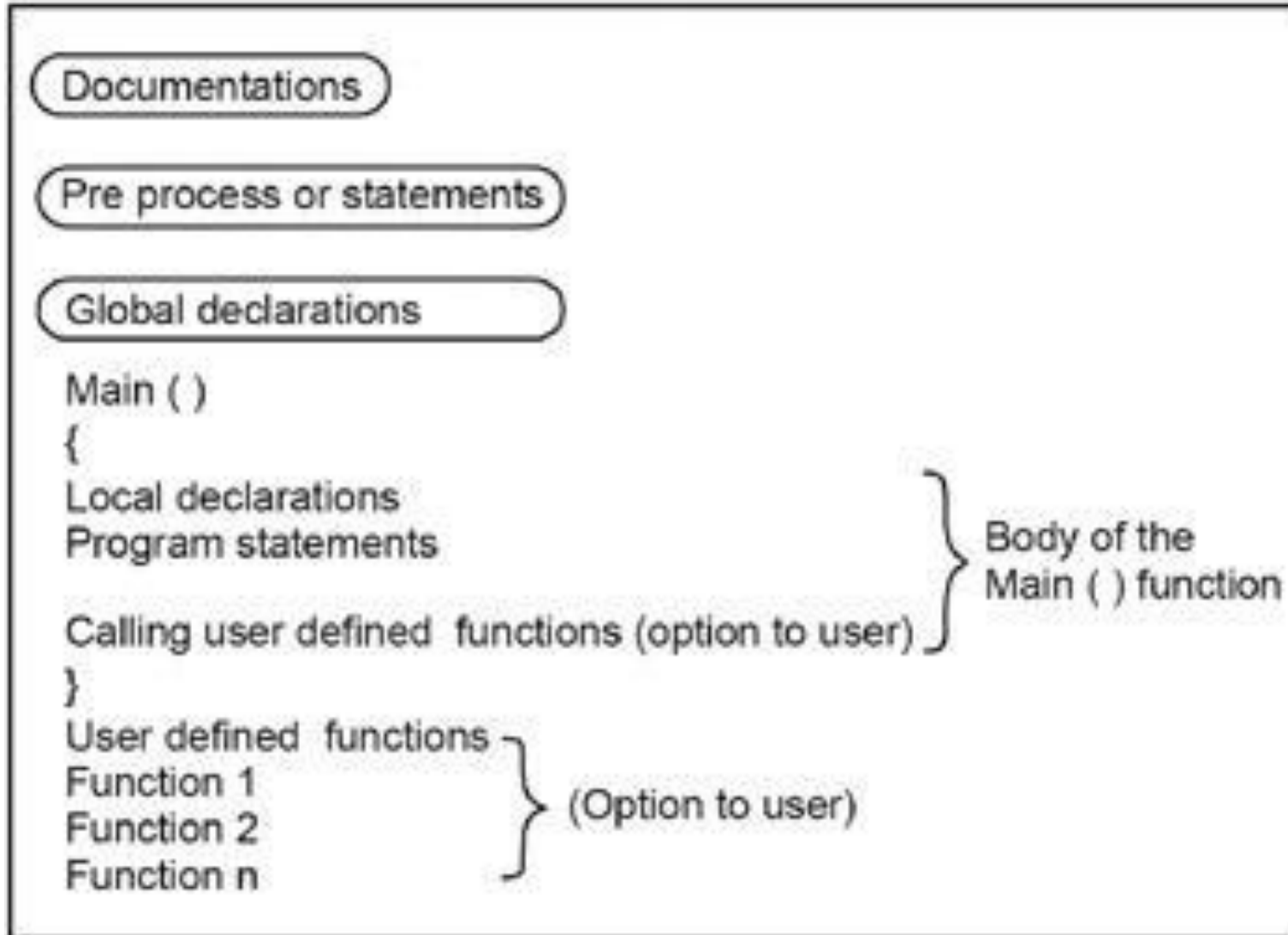
รายวิชา สธ 113 การออกแบบโปรแกรมทางธุรกิจเบื้องต้น

อ.อภิพงศ์ ปิงยศ

โครงสร้างโปรแกรม

- ▶ โครงสร้างการเขียนโปรแกรมด้วยภาษา C/C++ มี 5 ส่วน ดังนี้
 - ▶ 1. พรีโพรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)
 - ▶ 2. ส่วนประกาศ (Global declarations)
 - ▶ 3. ส่วนฟังก์ชันหลัก (The main() function)
 - ▶ 4. การสร้างฟังก์ชันและการใช้ฟังก์ชัน
 - ▶ 5. ส่วนอธิบายโปรแกรม (Program comments)

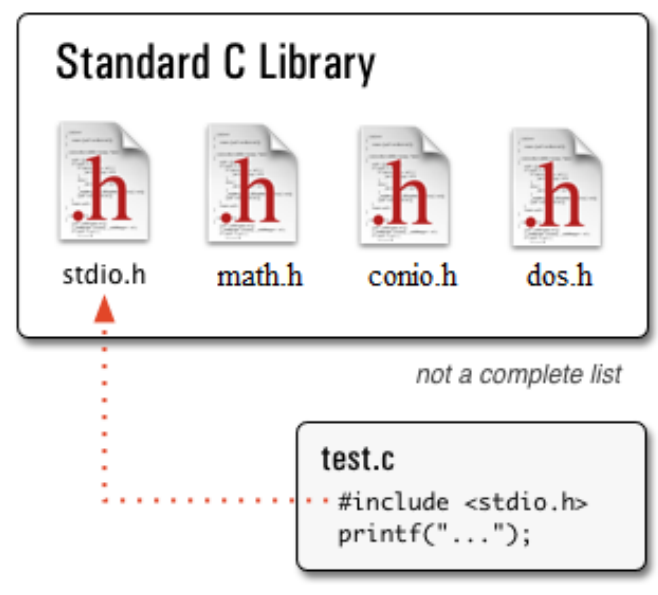
ตัวอย่างโครงสร้างโปรแกรม



โครงสร้างโปรแกรม:

1. พรีโพรเซสเซอร์ไดเรกทีฟ (Preprocessor directives)

- ▶ ใช้สำหรับเรียกไฟล์ที่โปรแกรมต้องการใช้งาน หรือกำหนดค่าต่างๆก่อนเป็นอันดับแรก เรียกอีกอย่างว่าเป็นส่วนหัวของโปรแกรมหรือ Header
- ▶ เริ่มต้นด้วยการใช้เครื่องหมายไดเรกทีฟ “#” และตามด้วยชื่อโปรแกรมหรือชื่อตัวแปรที่ต้องการใช้
- ▶ ไดเรกทีฟที่ใช้บ่อยๆได้แก่
 - ▶ `#include <ชื่อไฟล์>` เป็นการนำไฟล์อื่นเข้ามาคอมไพล์ร่วมด้วย
 - ▶ `#define NAME VALUE` เป็นการกำหนดค่าให้กับตัวแปร (ปกติจะใช้กับค่าคงที่)



โครงสร้างโปรแกรม:

2. ส่วนประกาศ (Global declarations)

- ▶ ต่อจากส่วนไคเร็กทีฟ จำเป็นต้องวางไว้นอกฟังก์ชันต่างๆ
- ▶ ใช้ประกาศตัวแปรตัวแปรหรือฟังก์ชันที่ต้องใช้ในโปรแกรม โดยทุกส่วนในโปรแกรมสามารถเรียกใช้ข้อมูลที่ประกาศไว้ในส่วนนี้ได้
- ▶ บางโปรแกรมอาจไม่ต้องมีก็ได้ ขึ้นอยู่กับรูปแบบของโปรแกรม

โครงสร้างโปรแกรม:

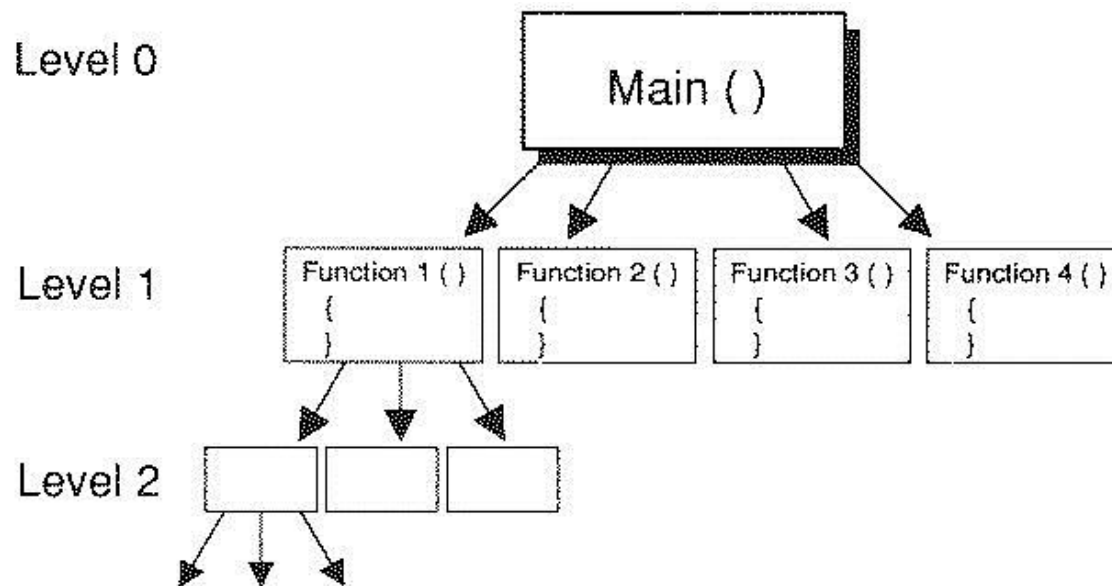
3. ส่วนฟังก์ชันหลัก (The main() function)

- ▶ เป็นส่วนที่ทุกโปรแกรมจำเป็นต้องมี ประกอบไปด้วยคำสั่งต่างๆที่จะให้โปรแกรมทำงาน
- ▶ แต่ละโปรแกรมจะต้องมีฟังก์ชันหลักเพียงฟังก์ชันเดียวเท่านั้น
- ▶ คอมไพเลอร์จะคอมไพล์ที่ฟังก์ชันนี้เป็นหลัก หากฟังก์ชันหลักมีการเชื่อมโยงไปหาฟังก์ชันอื่นๆอีก คอมไพเลอร์จึงจะไปคอมไพล์ฟังก์ชันที่ถูกอ้างนั้นต่อไป (รายละเอียดจะอยู่ในเรื่องฟังก์ชัน)

โครงสร้างโปรแกรม:

4. การสร้างฟังก์ชันและการใช้ฟังก์ชัน

- ▶ เป็นการเขียนฟังก์ชันย่อยๆขึ้นมาใช้ในโปรแกรม โดยฟังก์ชันเหล่านี้จำเป็นต้องถูกเรียกใช้งานผ่านฟังก์ชันหลัก (main() function)



โครงสร้างโปรแกรม:

5. ส่วนอธิบายโปรแกรม (Program comments)

- ▶ ใช้เขียนคอมเมนต์โปรแกรม เพื่ออธิบายการทำงานต่างๆ ทำให้ผู้ศึกษาโปรแกรมในภายหลังทำความเข้าใจโปรแกรมได้ง่ายขึ้น
- ▶ เมื่อคอมไพล์โปรแกรม ส่วนคอมเมนต์จะถูกข้ามไป
- ▶ การใช้คอมเมนต์มี 2 วิธีคือ
 - ▶ // ใช้คอมเมนต์ข้อความสั้นๆ ไม่เกินหนึ่งบรรทัด
 - ▶ /* ใช้คอมเมนต์ข้อความยาวๆ หลายบรรทัด และต้องปิดท้ายด้วย */ เสมอ

ตัวอย่างโปรแกรมภาษาซีเบื้องต้น

```
#include <stdio.h> /* needed for printf */  
  
int main()  
{  
    printf("Hello World\n");  
  
    return 1;  
}
```

ตัวแปร (Variables)

- ▶ หากต้องการให้โปรแกรมรับข้อมูลจากผู้ใช้มาเก็บไว้ หรือมีการคำนวณและเก็บผลลัพธ์ จะต้องสร้างตัวแปรสำหรับเก็บข้อมูลที่เป็นผลลัพธ์นั้น
- ▶ พิจารณาตัวอย่างต่อไปนี้

```
#include <stdio.h>
```

```
int feet, inch;
```

```
void main()
```

```
{
```

```
    feet = 6;
```

```
    inch = feet * 12;
```

```
    printf ("Height in inch is %d",inch);
```

```
}
```

การประกาศตัวแปร
แบบ Global
เป็นชนิดจำนวนเต็ม

ตัวแปร (Variables) [cont.]

- ▶ จากตัวอย่างจะเห็นได้ว่าการประกาศตัวแปรจะต้องมีการประกาศชนิดของตัวแปรนำหน้าชื่อตัวแปรก่อนเสมอ เช่น `int feet, inch;`
- ▶ ชนิดตัวแปรในที่นี้เป็นชนิดจำนวนเต็ม (Integer)
- ▶ หากเป็นตัวแปรชนิดเดียวกันสามารถใช้เครื่องหมาย Comma คั่นได้
- ▶ คำสั่ง `feet = 6` เป็นการสั่งให้นำค่า 6 ไปเก็บไว้ในตัวแปรชื่อ `feet`
- ▶ คำสั่ง `inch = feet * 12` เป็นการนำผลจากการคำนวณ `feet * 12` ไปเก็บไว้ในตัวแปรชื่อ `inch`
- ▶ `%d` เป็นการบอกให้นำค่าเลขฐานสิบในตัวแปร `inch` ไปแสดงผลทางหน้าจอ

การเก็บข้อมูลของตัวแปร

- ▶ ในการประกาศตัวแปรให้กับโปรแกรม ค่าในตัวแปรต่างๆจะถูกนำไปเก็บในหน่วยความจำเสมอ
- ▶ ค่าในตัวแปรและหน่วยความจำอาจมีการเปลี่ยนแปลงได้ โดยค่าเก่าจะหายไป และถูกแทนด้วยค่าใหม่ เช่น

```
int x = 6;  
x = 8;  
printf ("X is %d",x);
```

ค่าสุดท้ายของ
x คือ 8

การเก็บข้อมูลของตัวแปร [cont.]

- ▶ เราสามารถนำค่าในตัวแปรใส่ให้กับอีกตัวแปรหนึ่งได้ โดยใช้เครื่องหมาย “=” เช่นเดียวกับการกำหนดค่าให้ตัวแปรตามปกติ เช่น

```
int num1 = 5;  
int num2 = 14;  
num1 = num2;  
printf (“num1 is %d \n”,num1);  
printf (“num2 is %d \n”,num2 );
```

```
num1 is 14  
num2 is 14
```

การเก็บข้อมูลของตัวแปร [cont.]

- ▶ ในการกำหนดค่าให้ตัวแปร เราสามารถนำค่าตัวแปรเดิมมากระทำทางคณิตศาสตร์ได้ โดยตัวแปรทางซ้ายของเครื่องหมายเท่ากับ “=” จะเปลี่ยนไปตามการกระทำทางขวามือ เช่น

```
count = count + 1 //เป็นการเพิ่มค่าตัวแปร count ขึ้นหนึ่งค่า  
sum = sum + x //นำค่า sum บวกกับ x โดยค่าใน x จะไม่เปลี่ยน  
num = num * 3 //นำค่า num คูณกับ 3 แล้วเก็บค่าไว้ใน num เช่นเดิม
```