

CS 211 วิทยาการคอมพิวเตอร์ 1

โดย อาจารย์ภาณุวัฒน์ เมฆะ

สาขาวิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

ฟังก์ชัน (FUNCTION)

ฟังก์ชันคืออะไร

ฟังก์ชัน(function) คือ ส่วนของโปรแกรมที่ทำงานเสร็จสิ้นภายในตัวเอง มีลักษณะเหมือนกับโปรแกรมย่อย ที่รวมอยู่ในโปรแกรมหลักอีกทีหนึ่ง

หลักการ

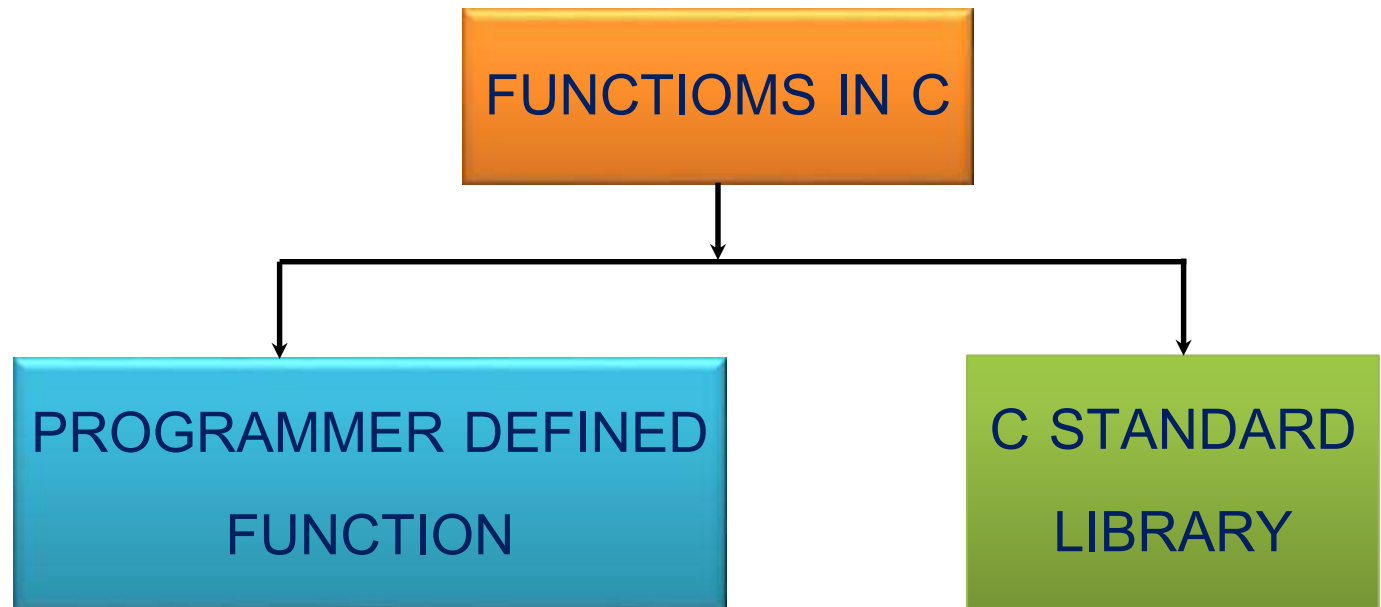
- แบ่งการทำงานออกเป็นส่วนเล็ก ๆ ที่ทำงานเสร็จสมบูรณ์ในตัวเอง
- ทดสอบและแก้ไขส่วนเล็ก ๆ นี้
- ประกอบส่วนเล็ก ๆ นี้ ขึ้นมาเป็นโปรแกรมขนาดใหญ่ที่สมบูรณ์ในขั้นตอนสุดท้าย

ทำไมต้องมีฟังก์ชัน

- เนื่องจากโปรแกรมเริ่มซับซ้อนมีขนาดใหญ่ เมื่อมีการกลับมาแก้ไข หรือต้องการปรับปรุง โปรแกรมเพิ่มเติมจะทำ ได้ยาก
- การเขียนโปรแกรมแบบฟังก์ชัน สามารถช่วย แก้ปัญหาความซับซ้อนได้

ฟังก์ชันในภาษา C

- ภาษา C แบ่งฟังก์ชันเป็น 2 แบบ
 1. ฟังก์ชันที่สร้างโดยผู้เขียนโปรแกรม
 2. ฟังก์ชันมาตรฐานใน C



ฟังก์ชันที่สร้างโดยผู้เขียนโปรแกรม

โครงสร้างโดยรวมของโปรแกรม 1

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

```
void func1() {  
    printf("1");  
}
```

```
void func2() {  
    func1();  
    printf("2");  
}
```

```
int main() {  
    func2();  
    printf("3");  
    return 0;  
}
```

ส่วนของการ include header file(S)

ฟังก์ชันย่อย (sub function)

ฟังก์ชันย่อย (sub function)

โปรแกรมหลัก

โครงสร้างที่ฟังก์ชันนิยามก่อนเรียกใช้
ไม่จำเป็นต้องมี ดัชนีแบบฟังก์ชัน

รูปแบบของฟังก์ชัน

การเขียนฟังก์ชันในภาษาซีสามารถแบ่งการเขียนออกเป็น 2 แบบหลักตามลักษณะการเขียน คือ

1. ฟังก์ชันที่ไม่มีการส่งค่ากลับเมื่อทำงานเสร็จ มี 2 แบบก็คือ

I. ฟังก์ชันที่มีการรับค่า แต่ไม่ส่งค่ากลับ

ฟังก์ชันนี้จะเป็นการส่งค่าพารามิเตอร์ parameter ให้แก่ฟังก์ชัน โดยไม่มีการส่งค่ากลับ และจะมีคำว่า void นำหน้าชื่อฟังก์ชัน

รูปแบบของฟังก์ชัน(ต่อ)

- ❖ รูปแบบการประกาศฟังก์ชันย่อย ใน (ฟังก์ชันหลัก หรือ main())

ชื่อฟังก์ชัน (ชนิดตัวแปร ตัวแปร ,...) ;

- ❖ รูปแบบของฟังก์ชันย่อย

void

ชื่อฟังก์ชัน

(ชนิดตัวแปร ตัวแปร ,...)

{

การประกาศตัวแปรภายในฟังก์ชัน ;

คำสั่ง;

}

รูปแบบของฟังก์ชัน(ต่อ)

- ฟังก์ชันที่มีการรับค่า แต่ไม่ส่งค่า

```
#include <stdio.h>
void Calc(int n) {
    int total;
    total = n*n*n;
    printf("Multiple is :
%d\n", total);
}
```

กำหนดฟังก์ชันชื่อ Calc
โดยมีการรับค่า parameter
มา 1 ตัวมีชนิดเป็น int เก็บไว้ที่ n

```
int main() {
    int number = 5;
    Calc(number);
    system("pause");
    return 0;
}
```

รูปแบบของฟังก์ชัน(ต่อ)

II. ฟังก์ชันที่ไม่มีการรับ/ส่งค่า

โดยฟังก์ชันไม่มีการรับค่าใดๆ เข้าไปในฟังก์ชัน และฟังก์ชันก็ไม่มีผลลัพธ์ใดๆ ส่งออกมา เป็นฟังก์ชันที่ ทำงานจบภายในฟังก์ชัน เช่น ฟังก์ชันการแสดงผล จะมีคำว่า void นำหน้าชื่อฟังก์ชัน

❖ รูปแบบการประกาศฟังก์ชันย่อย ใน (ฟังก์ชันหลัก หรือ main())

ชื่อฟังก์ชัน () ;

รูปแบบของฟังก์ชัน(ต่อ)

❖ รูปแบบของฟังก์ชัน

```
void ชื่อฟังก์ชัน ( ) {  
    การประกาศตัวแปรภายในฟังก์ชัน ;  
    คำสั่ง;  
}
```

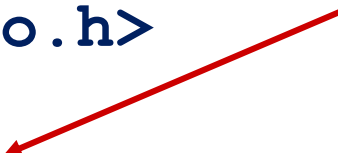
รูปแบบของฟังก์ชัน(ต่อ)

- ฟังก์ชันที่ไม่มีการรับ/ส่งค่า

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

```
void message ( ) {  
    printf("C Programming\n");  
    printf("is easy");  
}
```

กำหนดฟังก์ชันชื่อ message
โดยไม่มีการรับค่า และไม่มี
ผลลัพธ์ส่งกลับ



```
int main() {  
    message ( );  
    system("pause");  
    return 0;  
}
```

รูปแบบของฟังก์ชัน(ต่อ)

Ex1.1

เป็นฟังก์ชันแสดงรายชื่อภาพยนตร์ที่กำลังเข้าฉาย โดยให้
ผู้ใช้เลือกว่าต้องการตรวจสอบรายการภาพยนตร์หรือไม่

```
Do you want to check movies program? (y/n)y  
Now showing  
1.Spider man  
2.Harry Potter  
3.Resident Evil  
4.
```

รูปแบบของฟังก์ชัน(ต่อ)

Ex1.2

สร้างเมนูและฟังก์ชันในการคำนวณพื้นที่ของรูปร่างต่อไปนี้

วงกลม, สามเหลี่ยม, สี่เหลี่ยม

โดยให้เลือกรูปร่างที่ต้องการคำนวณพื้นที่จากนั้นให้นักศึกษาสร้างฟังก์ชันในการคำนวณพื้นที่ของแต่ละรูปร่าง

- วงกลม
- สามเหลี่ยม
- สี่เหลี่ยม

โดยให้เลือกรูปร่างที่ต้องการคำนวณพื้นที่จากนั้นให้นักศึกษาสร้างฟังก์ชันในการคำนวณพื้นที่ของแต่ละรูปร่าง

Menu

1.Circle

2.Triangle

3.Rectangle

Please select shape to calculate :

รูปแบบของฟังก์ชัน(ต่อ)

2. ฟังก์ชันที่มีการส่งค่ากลับเมื่อทำงานเสร็จ

ฟังก์ชันนี้ จะเป็นการส่งผ่านค่าพารามิเตอร์ (parameter) ไปกลับระหว่างฟังก์ชัน

❖ รูปแบบการประกาศฟังก์ชันย่อย ใน (ฟังก์ชันหลัก หรือ main())

ชื่อฟังก์ชัน (ตัวแปร ,...) ;

รูปแบบของฟังก์ชัน(ต่อ)

❖ รูปแบบฟังก์ชันที่มีการส่งค่ากลับ(return)

ชนิดข้อมูลที่คืนค่า

ชื่อฟังก์ชัน

(ชนิดตัวแปร ตัวแปร ,...)

{

การประกาศตัวแปรภายในฟังก์ชัน;

คำสั่ง;

return(ค่าข้อมูลที่ต้องการส่งค่ากลับ);

}

รูปแบบของฟังก์ชัน(ต่อ)

Ex2.1

ตัวอย่างโปรแกรมคำนวณเงินทอนลูกค้าโดยเมื่อลูกค้าซื้อสินค้าโปรแกรมจะทำการรับค่าราคาสินค้า และจำนวนเงินลูกค้า และสร้างฟังก์ชันคิดเงินทอนคืน

รูปแบบของฟังก์ชัน(ต่อ)

Ex2.2

ให้นักศึกษาเขียนโปรแกรมคำนวณแม่สูตรคูณ ของตัวเลขที่รับเข้ามาจากหน้าจอแสดงผล โดยการคำนวณให้แสดงเช่น

รับค่า 2 จากหน้าจอแสดงผลโปรแกรมคำนวณจะได้เป็น

```
Enter Number : 2
1 x 2 = 2
2 x 2 = 4
3 x 2 = 6
4 x 2 = 8
5 x 2 = 10
6 x 2 = 12
7 x 2 = 14
8 x 2 = 16
9 x 2 = 18
10 x 2 = 20
11 x 2 = 22
12 x 2 = 24
```

โดยใช้ฟังก์ชันมาช่วยในการคำนวณ ใครทำเสร็จแล้วให้ยกมือบอกพี่ ๆ ที่เอ้อผ่านแล้วให้มาเขียนรหัสตัวเองหน้าห้องเรียน

รูปแบบของฟังก์ชัน(ต่อ)

Ex2.3

เขียนโปรแกรมรับค่าสองค่าจากฟังก์ชัน main() และให้สร้างเมนูและฟังก์ชันในการคำนวณต่อไปนี

- ฟังก์ชัน Add() ในการบวกค่า
- ฟังก์ชัน Difference() ในการลบค่า
- ฟังก์ชัน Multiply() ในการคูณ
- ฟังก์ชัน Division() ในการหาร

โดยให้ส่งค่าผลลัพธ์การคำนวณกลับมาแสดงผลที่ฟังก์ชัน main() ตัวอย่างหน้าจอผลลัพธ์

```
Please Enter Number 1 : 25
Please Enter Number 2 : 20
Menu
1.Addition
2.Differnce
3.Multiply
4.Division
Please select Menu :
```

การประกาศโปรโตไทป์ของฟังก์ชัน

- การประกาศโปรโตไทป์เป็นสิ่งจำเป็นในภาษาซีเนื่องจากภาษาซีเป็นภาษาในลักษณะที่ต้องมีการประกาศฟังก์ชันก่อนจะเรียกใช้ฟังก์ชันนั้น
- หากต้องการย้ายฟังก์ชัน `main ()` ขึ้นไปไว้ด้านบน จะต้องมีการประกาศโปรโตไทป์ของฟังก์ชันที่ต้องการเรียกใช้ก่อนเสมอ

โครงสร้างโดยรวมของโปรแกรม 2

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

```
void func2();
```

```
void func1();
```

```
int main() {
```

```
    func2();
```

```
    printf("3");
```

```
    return 0;
```

```
}
```

```
void func2() {
```

```
    func1();
```

```
    printf("2");
```

```
}
```

```
void func1() {
```

```
    printf("1");
```

```
}
```

ส่วนของการ include header file(s)

การประกาศโปรโตไทป์

ฟังก์ชันหลัก

โครงสร้างที่ฟังก์ชันนิยามหลังการเรียกใช้
จำเป็นต้องมี ต้นแบบฟังก์ชัน

ฟังก์ชันย่อย (sub function)

ฟังก์ชันย่อย (sub function)

รูปแบบของ โปรโตไทป์ของฟังก์ชัน

ชนิดข้อมูล ชื่อฟังก์ชัน (ชนิดของข้อมูล);

void ชื่อฟังก์ชัน (ชนิดของข้อมูล);

หรือ

ชนิดข้อมูล ชื่อฟังก์ชัน ();

void ชื่อฟังก์ชัน ();

****** ต้องประกาศโปรโตไทป์ไว้บนสุดเสมอ

รูปแบบของ โปรโตไทป์ของฟังก์ชัน

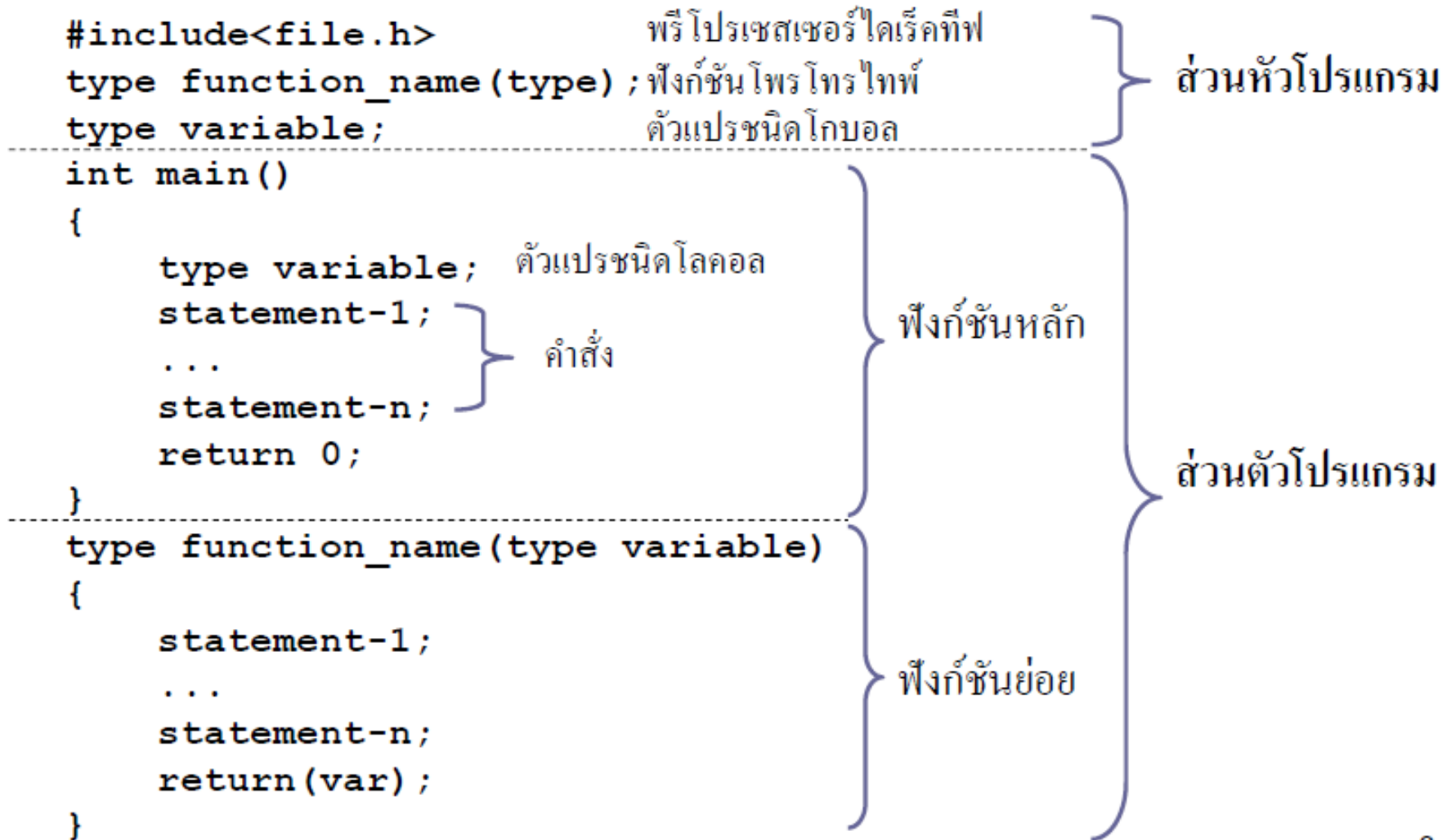
ตัวอย่าง การประกาศ
โปรโตไทป์ของฟังก์ชัน

*** สังเกต เมื่อมีการประกาศ
โปรโตไทป์แล้ว ฟังก์ชันหลัก
จะอยู่บนฟังก์ชันย่อย

```
#include <stdio.h>
int square(int);
int main()
{
    int a=2,b;
    b = square(a);
    printf("b = %d \n",b);
    return 0;
}
```

```
int square(int x)
{
    int y;
    y = x*x;
    return y;
}
```

โครงสร้างโปรแกรมภาษาซี

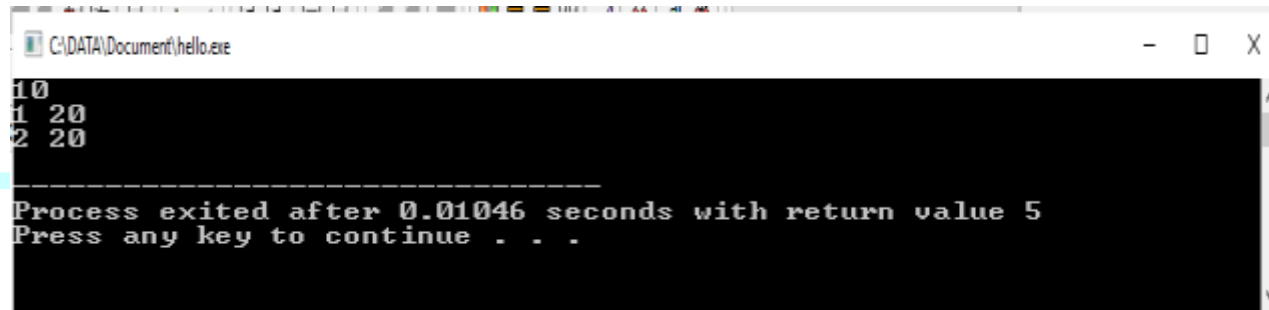


ตัวแปร Global และ Local

```
1 #include<stdio.h>
2 int c=10;
3
4 void function1()
5 {
6     int a=1;
7     printf("%d %d\n",a,c);
8 }
9
10 void function2()
11 {
12     int a=2;
13     printf("%d %d\n",a,c);
14 }
15
16
17 void main()
18 { printf("%d\n",c);
19   c=20;
20   function1();
21   function2();
22 }
```

a ใน function1 เป็นคนละตัวกับ a ใน function2

ตัวแปร C เป็นตัวแปรตัวเดียวกันในทุกส่วนของโปรแกรม



รูปแบบของ โปรโตไทป์ของฟังก์ชัน

Ex3

จงเขียนโปรแกรมการบวกเลขจำนวนจริง 2 จำนวนที่รับ
จากผู้ใช้งาน (function prototype)

```
Input real value : 4
Input real value : 6
Result of sum is : 10.001
```

QUIZ

เขียนโปรแกรมเพื่อทำ หน้าที่ยาผลลัพท์ของการคำนวณจากสมการ

$x = a^2 \left(c + \frac{b}{2a} \right)$ โดยเขียนฟังก์ชันเพื่อรับข้อมูลเลขจำนวน

เต็ม a b และ c ฟังก์ชันเพื่อคำนวณผลลัพท์จากสมการและฟังก์ชัน
เพื่อแสดงผลลัพท์

(โดยใช้รูปแบบของ โปรโตไทป์ฟังก์ชัน)

เฉลย