

Computer Engineering
วิศวกรรมคอมพิวเตอร์



บทที่ 7 พื้นฐานเบื้องต้นและ ขอบเขตของตัวแปร

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



วัตถุประสงค์

01006012 Computer Programming

- เข้าใจหลักการของฟังก์ชัน
- สามารถใช้งานฟังก์ชันสำเร็จรูปได้
- เข้าใจตัวแปรชนิด โกลบออลและ โลคัล
- เข้าใจการส่งค่าระหว่างฟังก์ชันแบบต่างๆ
- สามารถสร้างฟังก์ชันอย่างง่ายได้
- สร้างฟังก์ชันที่มีการรับส่งค่าระหว่างฟังก์ชันได้



7.6 พังก์ชัน

01006012 Computer Programming

- พังก์ชัน คือ ชุดของการทำงาน ที่ถูกเขียนขึ้นให้โปรแกรมเมอร์สามารถเรียกใช้งานได้โดยง่าย

** พังก์ชัน คือ ชุดของการทำงาน

** พังก์ชัน ถูกเรียกใช้งานได้

ข้อดีของฟังก์ชัน

01006012 Computer Programming

- ทำให้โปรแกรมเมอร์สามารถพัฒนาโปรแกรมได้โดยง่าย โดยโปรแกรมเมอร์ไม่จำเป็นต้องทราบว่าการทำงานของฟังก์ชันทำงานอย่างไร ทราบเพียงผลลัพธ์ของการทำงานเท่านั้น
- โปรแกรมเมอร์สามารถเขียนโปรแกรมให้มีการทำงานที่ซับซ้อนได้โดยไม่จำเป็นต้องเขียนโปรแกรมส่วนที่ซับซ้อนนั้นหลายๆ ครั้ง
- โปรแกรมเมอร์สามารถออกแบบแบบฟังก์ชันได้ตามความต้องการของโปรแกรมเมอร์



ฟังก์ชัน (Function)แบ่งออกเป็น 2 ประเภท

01006012 Computer Programming

1. ฟังก์ชันไลบรารีมาตรฐาน (Standard Library function)
2. ฟังก์ชันที่สร้างขึ้นเอง (User Defined function)

ฟังก์ชันไอลบรารีมาตรฐาน (Standard Library Function)

01006012 Computer Programming

- ฟังก์ชันไอลบรารีมาตรฐาน (Standard Library Function) เป็นฟังก์ชันที่มีอยู่แล้วเก็บไว้ใน Library ในการใช้งานต้อง include directives ก่อน
- directive คือสารบัญของกลุ่มฟังก์ชัน เช่น stdio.h , conio.h , string.h , math.h เป็นต้น
- การ include directives จะเป็นเหมือนการประกาศให้กับ compiler ทราบว่าจะใช้คำสั่ง ในกลุ่มของ directive นั้นๆ เช่น การใช้คำสั่ง sin() ซึ่งอยู่ใน math.h จะต้องมีบรรทัด include math.h เสมอ ดังตัวอย่าง



การเรียกใช้ Standard Library Function

01006012 Computer Programming

- การเรียกใช้ Standard Library Function มีขั้นตอนดังนี้
 - ทราบว่าฟังก์ชันที่ต้องการใช้งานอยู่ใน header file (ไฟล์ .h) ได
 - Include header file นั้น
 - เรียกใช้ฟังก์ชันในโปรแกรม



ตัวอย่างไลบรารีฟังก์ชัน

01006012 Computer Programming

- ฟังก์ชันการคำนวณทางคณิตศาสตร์
 - ไฟล์ header => math.h
- ฟังก์ชันสำหรับอักขระและข้อความ
 - ไฟล์ header => string.h
 - ctype.h



ฟังก์ชันการคำนวณทางคณิตศาสตร์

01006012 Computer Programming

#include<math.h>

```
sin(var);  
cos(var);  
tan(var);  
sqrt(var) ;// รากที่ 2  
pow(var1,var2) ;// var1 ยกกำลัง var2  
log(var) ;// logฐาน e  
log10(var) ;// logฐาน 10  
exp(var) ;// e ยกกำลัง  
fabs(var) ;// ค่าสัมบูรณ์แบบfloat
```

ตัวอย่างการใช้งานฟังก์ชันการคำนวณทางคณิตศาสตร์



01006012 Computer Programming

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#define PI 3.14
int main()
{
    float deg,rad;
    printf ("Enter Degree : ");
    scanf ("%f",&deg);
    rad = deg * PI / 180;
    printf ("sin(% .2f) = % .3f\n",deg,sin(rad));
    printf ("cos(% .2f) = % .3f\n",deg,cos(rad));
    printf ("tan(% .2f) = % .3f\n",deg,tan(rad));
    return 0;
}
```

ฟังก์ชันสำหรับข้อความ

01006012 Computer Programming

#include<string.h>

```
strcpy(str1, str2) ;  
strcat(dest1, src2) ;  
strcmp(dest1, src2) ;  
strcmpi(str1, str2) ;  
strlen(str) ;
```

#include<ctype.h>

```
tolower(ch) ;  
toupper(ch) ;
```

ฟังก์ชันที่สร้างขึ้นเอง (User-defined Function)



01006012 Computer Programming

- เนื่องจาก Standard Library Function ทั้งหมด เป็นฟังก์ชันมาตรฐานที่มีเฉพาะการทำงานพื้นฐานต่างๆ เท่านั้น
- หากต้องการฟังก์ชันที่มีการทำงานเฉพาะกิจ โปรแกรมเมอร์ต้องเขียนฟังก์ชันขึ้นมาเอง

ฟังก์ชันที่สร้างขึ้นเอง (User-defined Function)



01006012 Computer Programming

การเขียนโปรแกรมโดยมีฟังก์ชันที่สร้างขึ้นเอง มี 2 รูปแบบ

- สร้างฟังก์ชัน ก่อน ฟังก์ชันหลัก

ฟังก์ชันหลักสามารถเรียกใช้งานฟังก์ชันที่สร้างขึ้นได้

- สร้างฟังก์ชัน หลัง ฟังก์ชันหลัก

ต้องประกาศ Function Prototype ก่อนเพื่อให้ฟังก์ชันหลักรู้ว่ามีฟังก์ชันที่สร้างขึ้น

ໂປຣໂຕໄທປ່ານີ້ພັກໍ່ຂັ້ນ (Function Prototype)



01006012 Computer Programming

ເປັນການປະກາສການໃຊ້ງານພຶກໍ່ຫຸ້ນທີ່ອຢູ່ໜັງ main()

```
type function_name(type-1, type-2, . . . , type-n);
```

function name กือ ชื่อฟังก์ชันที่จะสร้างขึ้น

type-n គីវ ចនិតុសងខ្មែរមូលទៀតេសំសៀង ឲ្យដើរកំបាន



สร้างฟังก์ชันก่อนฟังก์ชันหลัก

01006012 Computer Programming

```
#include<file.h>
type variable
type function_name(type variable)
{
    statement-1;
    ...
    statement-n;
    return(var);
}

int main()
{
    type variable;
    statement-1;
    ...
    statement-n;
    return 0;
}
```



สร้างฟังก์ชันหลังฟังก์ชันหลัก

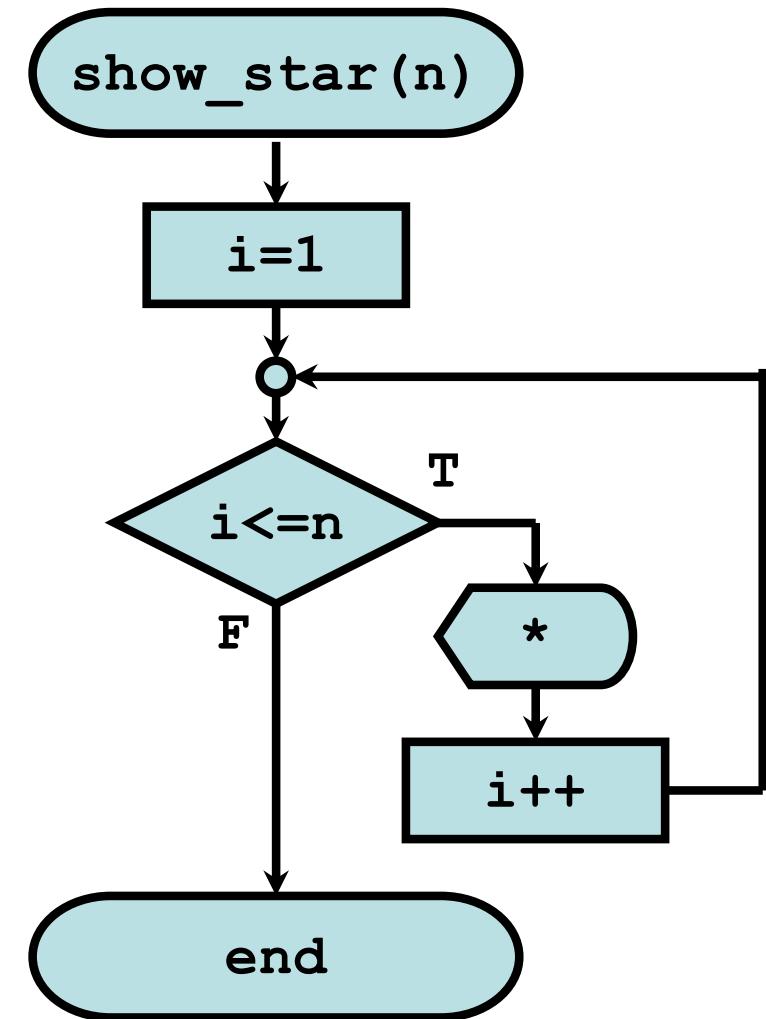
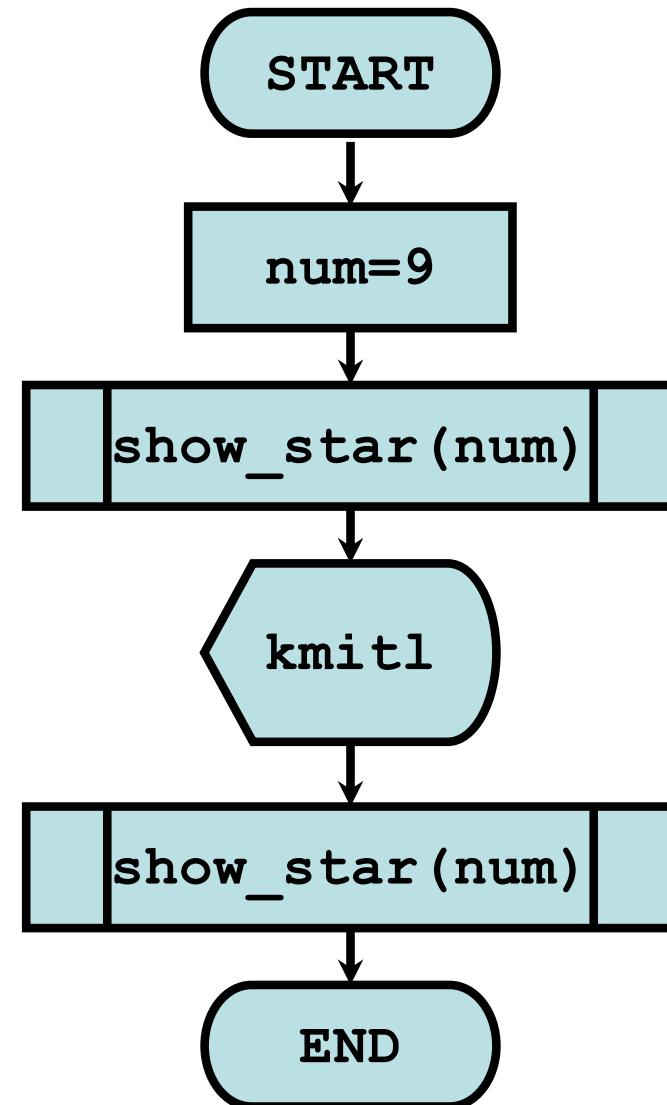
01006012 Computer Programming

```
#include<file.h>
type function_name(type);
type variable
int main()
{
    type variable;
    statement-1;
    ...
    statement-n;
    return 0;
}
type function_name(type variable)
{
    statement-1;
    ...
    statement-n;
    return(var);
}
```

ตัวอย่าง | ผังงาน & ผลลัพธ์

01006012 Computer Programming

```
*****
* kmitl *
*****
```



ตัวอย่าง | โปรแกรม

01006012 Computer Programming

```
#include<stdio.h>
#include<conio.h>
void show_star (int n)
{
    int i;
    for (i=1;i<=n;i++)
        putchar('*');
}
int main()
{
    int num=9;
    show_star(num);
    printf ("\n* kmitl *\n");
    show_star(num);
    return 0;
}
```

```
#include<stdio.h>
#include<conio.h>
void show_star (int) ;
int main()
{
    int num=9;
    show_star(num);
    printf ("\n* kmitl *\n");
    show_star(num);
    return 0;
}
void show_star (int n)
{
    int i;
    for (i=1;i<=n;i++)
        putchar('*');
}
```



ประเภทของฟังก์ชันที่สร้างขึ้นเอง

01006012 Computer Programming

- ฟังก์ชันที่ไม่มีการรับส่งค่า
- ฟังก์ชันที่มีการรับค่า แต่ไม่ส่งค่ากลับ
- ฟังก์ชันที่มีการรับค่า และมีการส่งค่ากลับ
- ฟังก์ชันที่ไม่มีการรับค่า แต่มีการส่งค่ากลับ

** สามารถแยกประเภทได้โดยดูจาก function prototype

ฟังก์ชันที่ไม่มีการรับส่งค่า

01006012 Computer Programming

- เป็นฟังก์ชันที่ไม่มีการรับค่าเข้ามาในฟังก์ชัน และไม่มีการส่งค่ากลับออกไปจากฟังก์ชัน

```
#include<stdio.h>
#include<conio.h>
void function_name(void);
int main()
{
    ...
    function_name();
    ...
    return 0;
}
```

```
void function_name()
{
    statement-1;
    statement-2;
    ...
    statement-n;
}
```

ตัวอย่าง พังก์ชันที่ไม่มีการรับส่งค่า

01006012 Computer Programming

```
#include<stdio.h>

void PrintHello(void);

int main()
{
    PrintHello();
    printf("Hello,in function main.\n");
    PrintHello();
    return 0;
}

void PrintHello(void)
{
    printf("Hello,in function PrintHello ..\n\n");
}
```



ฟังก์ชันที่มีการรับค่า แต่ไม่ส่งค่ากลับ

01006012 Computer Programming

- เป็นฟังก์ชันที่มีการรับค่าเข้ามาในฟังก์ชัน แต่ไม่มีการส่งค่ากลับออกไปจากฟังก์ชัน

```
#include<stdio.h>
#include<conio.h>
void func_name(type);
int main()
{
    ...
    func_name(varX);
    ...
    return 0;
}
```

```
void func_name(type varY)
{
    statement-1;
    statement-2;
    ...
    statement-n;
}
```

ตัวอย่างฟังก์ชันที่มีการรับค่า แต่ไม่ส่งค่ากลับ



```
#include<stdio.h>
void PrintHello( int );
int main()
{
    int n;
    printf("input number of hello : ");
    scanf("%d",&n);
    PrintHello( n );
    return 0;
}
void PrintHello( int i )
{
    int count;
    for ( count=1; count<=i ; count++ )
        printf("%d HELLO\n",count);
}
```

ฟังก์ชันที่มีการรับค่า และมีการส่งค่ากลับ

01006012 Computer Programming

- เป็นฟังก์ชันที่มีการรับค่าเข้ามาในฟังก์ชัน และมีการส่งค่ากลับออกไปจากฟังก์ชัน

```
#include<stdio.h>
#include<conio.h>
type func_name(type);
int main()
{
    ...
    var = func_name(varX);
    ...
    return 0;
}
```

```
type func_name(type varY)
{
    statement-1;
    statement-2;
    ...
    statement-n;
    return (varZ);
}
```

ตัวอย่าง พึงชั่นที่มีรับค่า มีส่งค่า



```
#include<stdio.h>

float CircleArea( int );
int main()
{
    int radius;
    printf("input radius : ");
    scanf("%d",&radius);
    printf(" Circle area = %f\n",CircleArea(radius));
    return 0;
}

float CircleArea( int rad )
{
    float answer=0;
    answer = 22.0/7.0*rad*rad ;
    return answer;
}
```

Computer Programming

ฟังก์ชันที่ไม่มีการรับค่า แต่มีการส่งค่ากลับ

01006012 Computer Programming

- เป็นฟังก์ชันที่ไม่มีการรับค่าเข้ามาในฟังก์ชัน แต่มีการส่งค่ากลับออกไปจากฟังก์ชัน

```
#include<stdio.h>
type func_name(void) ;
int main()
{
    ...
    var = func_name() ;
    ...
    return 0 ;
}
```

```
type func_name()
{
    statement-1;
    statement-2;
    ...
    statement-n;
    return (varZ) ;
}
```

ตัวอย่างฟังก์ชันที่ไม่มีการรับค่า แต่มีการส่งค่ากลับ



```
#include<stdio.h>

int RoundInput( void );
int main()
{
    int i,round;
    round = RoundInput();
    for( i=1; i<=round;i++)
        printf("hello # %d\n",i);
    return 0;
}

int RoundInput(void)
{
    int answer;
    printf( "Please input number of hello : ");
    scanf( "%d", &answer );
    return answer;
}
```

ตัวแปรและขอบเขตของการใช้งานสำหรับฟังก์ชัน



01006012 Computer Programming

- **ตัวแปร global**

เป็นตัวแปรที่ฟังก์ชันได้สามารถเรียกใช้ได้โดยจะประกาศสร้างตัวแปรต่อจาก พรี โพรແแซສເเซອ່ໄດເຮັກທີ່ພ

- **ตัวแปร local**

เป็นตัวแปรที่สามารถเรียกใช้ได้ภายในเฉพาะฟังก์ชันที่ประกาศสร้างตัวแปรนั้น โดยจะประกาศสร้างตัวแปรภายในแต่ละฟังก์ชัน

```
#include<stdio.h>
int num1; // num1 is global variable
void test(void); /*Function Prototype*/
int main()
{
    num1 = 19; // no num1 declaration
    printf ("line1 (main) : num1 = %d\n",num1);
    test();
    printf ("line2 (main) : num1 = %d\n",num1);
    return 0;
}
void test()
{
    num1 = 26; // no num1 declaration
    printf ("line1 (test) : num1 = %d\n",num1);
}
```

```
line1 (main) : num1 = 19
line1 (test) : num1 = 26
line2 (main) : num1 = 26
```

```
#include<stdio.h>
void test(void); /*Function Prototype*/
int main()
{
    int      num1; // local num1 in main()
    num1 = 19;
    printf ("line1 (main) : num1 = %d\n",num1);
    test();
    printf ("line2 (main) : num1 = %d\n",num1);
    return 0;
}
void test()
{
    int      num1; // local num1 in test()
    num1 = 26;
    printf ("line1 (test) : num1 = %d\n",num1);
}
```

```
line1 (main) : num1 = 19
line1 (test) : num1 = 26
line2 (main) : num1 = 19
```



จงแสดง output จากโปรแกรมต่อไปนี้

01006012 Computer Programming

```
#include<stdio.h>
void test(void); /*Function Prototype*/
int num1; //global num1
int main()
{
    int      num1; // local num1 in main()
    num1 = 19;
    printf ("line1 (main) : num1 = %d\n",num1);
    test();
    printf ("line2 (main) : num1 = %d\n",num1);
    return 0;
}
void test()
{
    num1 = 26;
    printf ("line1 (test) : num1 = %d\n",num1);
}
```

การส่งค่าตัวแปร (pass by value & pass by reference)

01006012 Computer Programming

การส่งค่าตัวแปร มีสองชนิดคือ

- การส่งค่าที่เก็บอยู่ในตัวแปรให้กับฟังก์ชัน (pass by value)
- การส่งค่า Address ของตัวแปรให้กับฟังก์ชัน (pass by reference)

**สำหรับในบทนี้จะกล่าวถึงเฉพาะ pass by value

Pass by Value

01006012 Computer Programming

- เป็นการส่งค่าที่เก็บอยู่ในตัวแปรเข้าสู่ฟังก์ชัน
- การเปลี่ยนแปลงค่าต่างๆ ของพารามิเตอร์จะ **ไม่เปลี่ยนแปลงค่า** ของตัวแปรในโปรแกรมหลัก
- ลักษณะของ Pass by Value Function จะ **ไม่ส่งค่า Address** (**หน้า** **ตัวแปร**จะ **ไม่มีเครื่องหมาย ***)
- ตัวอย่าง
 - int add(int a, int b)
 - void draw_line(int count)

Pass by Value Function

01006012 Computer Programming

```
void func(int va)
{
    va = va+1;
    printf ("In function la = %d\n",va);

}

int main()
{
    int x;
    x = 10;
    printf ("Before call function x = %d\n",x);
    func(x);
    printf ("After call function x = %d\n",x);
    return 0;
}
```

Output

01006012 Computer Programming

```
Before call function x = 10
In function la = 11
After call function x = 10
```



ตัวอย่างฟังก์ชัน

01006012 Computer Programming

```
#include <stdio.h>

int      findmax(int x,int y)
{
    if (x>y)
        return x ;
    else
        return y ;
}

void main(void)
{
    int a,b,d;
    scanf("%d %d", &a, &b) ;
    d = findmax(a,b) ;
    printf ("Max of %d,%d is %d",a,b,d);
}
```

จงเขียนโปรแกรมเพื่อรับตัวเลขจำนวนเต็มเข้าทาง
คีย์บอร์ด 2 จำนวน ให้โปรแกรมหาค่าสูงสุดโดย
ใช้ฟังก์ชัน **findmax(int,int)**

Output

```
10 2
Max of 10,2 is 10
```

คำสั่งท้ายบท

01006012 Computer Programming

1. จงเขียนโปรแกรมหาพื้นที่ของสามเหลี่ยมใด ๆ โดยรับค่าด้านทั้งสาม(a,b,c)ทางแป้นพิมพ์ จากสูตร

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = \frac{a + b + c}{2}$$

2. จงเขียนโปรแกรมแลกเปลี่ยนเงินสกุลสหราชอาณาจักรไทย โดยรับค่าเงินдолลาร์จากแป้นพิมพ์แล้วแสดงผลเป็นบาท โดยใช้ฟังก์ชัน D2B(int) เพื่อทำการแปลงдолลาร์ให้เป็นเงินบาท โดยกำหนดให้ 1 долลาร์มีค่า 31.25 บาท

3. จากส่วนของโปรแกรมต่อไปนี้ เป็นโปรแกรมตรวจสอบตัวเลขที่รับเข้ามาทางแป้นพิมพ์ ว่าเป็นจำนวนเฉพาะหรือไม่ จงเติมส่วนที่หายไปให้ครบ

---→



```
#include<stdio.h>
int main()
{
    int N;
    printf("Input N: ");
    scanf("%d", &N);
    if ( IsPrime(N) )
        printf("%d is Prime number ", N );
    else
        printf("%d is not Prime number ", N );

    return 0;
}
```