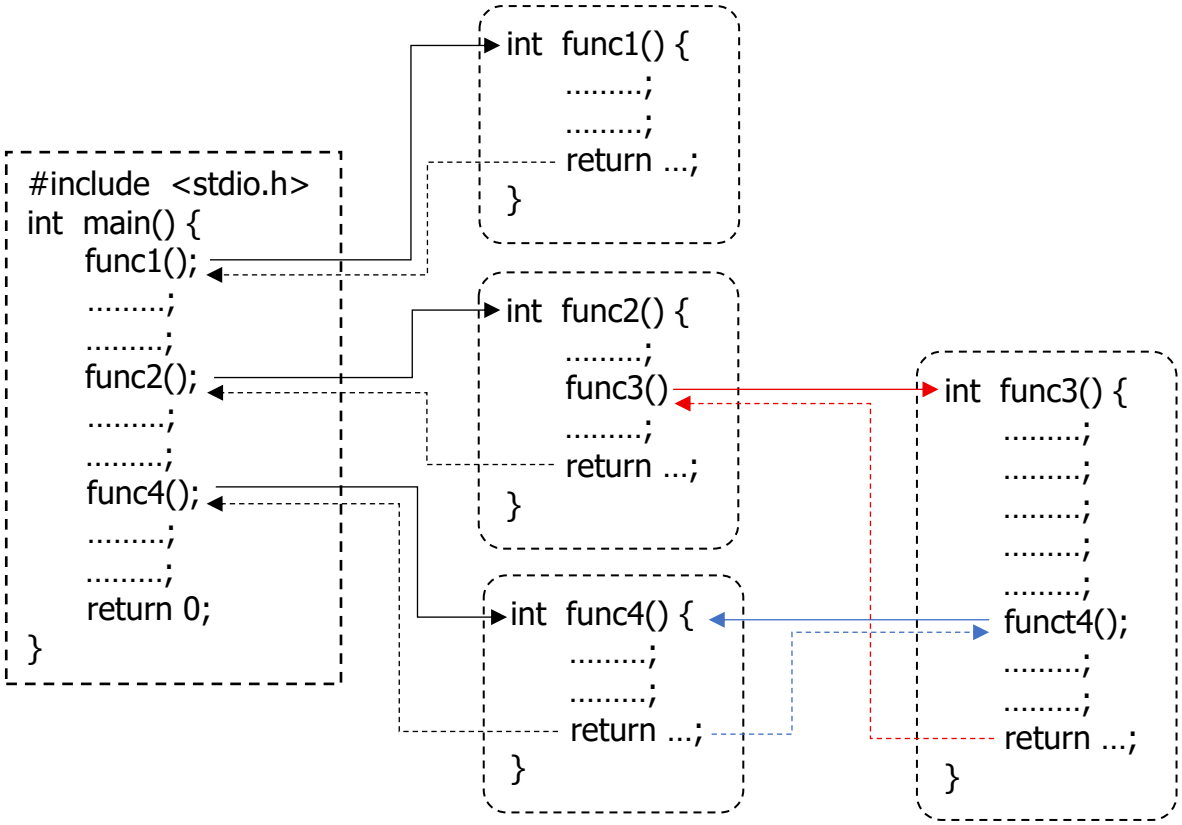
หน่วยที่ 5 ฟังก์ชัน (Function)

การเขียนโปรแกรมคอมพิวเตอร์นั้นสามารถสร้างคำใหม่ขึ้นมาได้ และสามารถเรียกใช้ได้เมื่อต้องการ โดยคำสั่งใหม่ที่สร้างขึ้นจะประกอบด้วยฟังก์ชันต่าง ๆ รวมกันอยู่ ถ้าหากคำใหม่ที่สร้างขึ้นนี้ไม่มีการคืนค่าออกมา การทำงานจะทำงานเป็นโปรแกรมย่อย ถ้ามีการคืนค้ากลับออกมาจะทำงานเป็นฟังก์ชัน ถ้าหากผู้เขียนโปรแกรมทำความเข้าใจวิธีการสร้างคำใหม่หรือการสร้างฟังก์ชันนี้จะทำให้โปรแกรมทำงานได้มีประสิทธิภาพมากยิ่งขึ้น

ฟังก์ชัน (Function) เป็นกลุ่มคำสั่งที่สร้างขึ้นมาเพื่อให้การทำงานอย่างใดอย่างหนึ่ง สามารถเรียกขึ้นมาใช้ได้หลายครั้ง ทำให้โปรแกรมดูง่ายขึ้น สามารปรับปรุงแก้ไขโปรแกรมได้ง่ายขึ้น ในการเขียนโปรแกรมด้วยภาษาซีจะต้องมีฟังก์ชัน main() เสมอ และถ้าต้องการให้โปรแกรมทำงานใด ๆ ฟังก์ชัน main() จะไปเรียกใช้ฟังก์ชันอื่น ๆ ให้โปรแกรมทำงาน ในภาษาซีจะมีไลบรารีสำหรับเก็บค่าฟังก์ชันต่าง ๆ ให้เราเลือกใช้งาน เช่น ฟังก์ชัน printf() จะเก็บไว้ใน stdio.h ซึ่งเคยทราบมาแล้ว



ฟังก์ชันในภาษาซี แบ่งออกได้เป็น 2 ประเภท

1. **ฟังก์ชันมาตรฐาน (Standard Functions)** เป็นฟังก์ชันที่มีอยู่แล้วและเก็บไว้ในไลบรารี เช่น ฟังก์ชันแสดงผล ฟังก์ชันรับข้อมูล ฟังก์ชันรับตัวอักขระ ฟังก์ชันลบจอภาพ เป็นต้น ในการใช้งานเราต้องเรียกใช้ #include เพื่อเรียกไฟล์ส่วนหัวของโปรแกรมขึ้นมาก่อน จึงสามารถใช้งานฟังก์ชันนั้น ๆ ได้ เช่น #include <stdio.h> หรือ #include <math.h>

2. **ฟังก์ชันที่ผู้เขียนโปรแกรมสร้างขึ้น (User-Defined Functions)** เป็นฟังก์ชันหรือโปรแกรมย่อยที่ผู้ใช้สร้างขึ้นมาใช้ในการเขียนโปรแกรม เพื่อทำงานใดงานหนึ่ง ซึ่งทำให้โปรแกรมดูง่ายขึ้นและลดความซ้ำซ้อนลง มี 4 รูปแบบ

   2.1 **No arguments, no return value** ฟังก์ชันนี้ไม่รับข้อมูลอินพุต และไม่ส่งคืนผลลัพธ์ใด ๆ

   2.2 **Arguments, no return value** ฟังก์ชันนี้รับข้อมูลอินพุต แต่ไม่ส่งคืนสิ่งใดเลย

   2.3 **No arguments, return value** ฟังก์ชันนี้ไม่รับข้อมุลอินพุต แต่ส่งคืนผลลัพธ์

   2.4 **Arguments and return value** ฟังก์ชันนี้รับข้อมูลอินพุต และส่งคืนผลลัพธ์

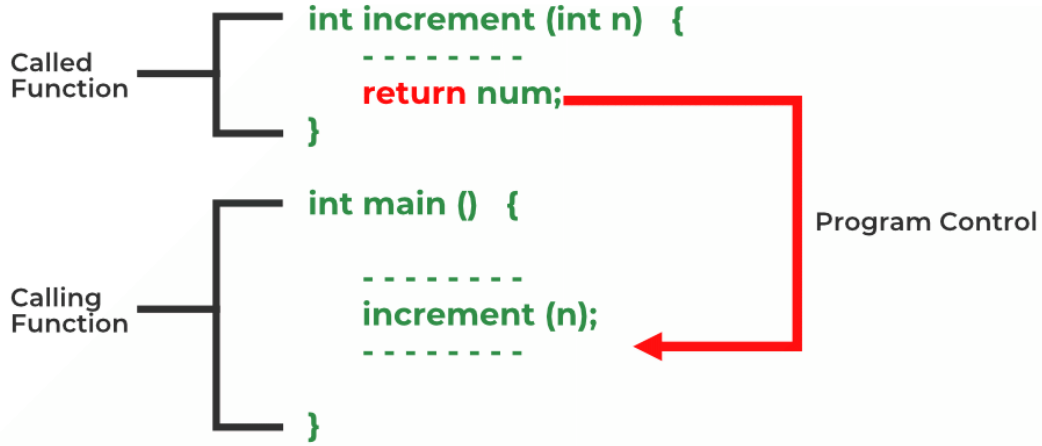
รูปแบบ

```
return_type  function_name(DataType  parameter1, DataType  parameter1, ... )
{
    .....;
    [return ...];
}
```

return_type	ประเภทหรือชนิดของข้อมูลที่ส่งกลับ เช่น int, float, chat, void เป็นต้น
function_name	ชื่อฟังก์ชัน ตั้งตามกฎการตั้งชื่อ
DataType	ประเภทข้อมูลของพารามิเตอร์
parameter	ตัวแปรหรือพารามิเตอร์ที่ใช้รับค่าผ่านเข้ามาในฟังก์ชัน มีหรือไม่มีก็ได้ ขึ้นอยู่กับการทำงานของฟังก์ชัน เช่น ชนิดข้อมูล พารามิเตอร์1, ชนิดข้อมูล พารามิเตอร์2, ...
return	ค่าที่ส่งกลับ มีหรือไม่มีก็ได้ ถ้ามีการส่งค่ากลับจะต้องเขียนค่าที่ส่งกลับตามหลัง return

Example

```
// function declaration
    int add(int a, int b);
    float  square(int a, int b);
    void  line();
    void sub(int x, int y);
```



2.1 **No arguments, no return value**: ฟังก์ชันนี้ไม่รับข้อมูลอินพุต และไม่ส่งคืนผลลัพธ์ใด ๆ

Ex01. เขียนโปรแกรมฟังก์ชันแสดงผลบวกและผลลบของจำนวนเต็มสองจำนวน โดยไม่มีการส่งค่าใด ๆ

```
#include  <stdio.h>
void add(){
      int  a=100, b=200;
      int sum = a + b;
      printf("%d + %d = %d\n", a, b, sum);
}

void sub(){
      int  x=500, y=300;
      int result = x – y;
      printf("%d – %d = %d\n", x, y, result);
}

int  main() {
      add();
      sub();
      return 0;
}
```

ผลการรันโปรแกรม (Output)

```
100 + 200 = 300
500 – 300 = 200
```

2.2 **Arguments, no return value:** ฟังก์ชันนี้รับข้อมูลอินพุต แต่ไม่ส่งคืนสิ่งใดเลย

Ex02. เขียนโปรแกรมฟังก์ชันแสดงผลบวกและผลลบของจำนวนเต็มสองจำนวน จากการส่งข้อมูลจากฟังก์ชัน main()

```
#include  <stdio.h>
void add(int m, int n){
      int sum = m + n;
      printf("%d + %d = %d\n", m, n, sum);
}

void sub(int x, int y){
      int result = x – y;
      printf("%d – %d = %d\n", x, y, result);
}

int  main() {
      int  a=100, b=200;
      int  x=500, y=300;
      add(a, b);
      sub(x, y);
      return 0;
}
```

ผลการรันโปรแกรม (Output)

```
100 + 200 = 300
500 – 300 = 200
```

2.3 No arguments, return value: ฟังก์ชันนี้ไม่รับข้อมูลอินพุต แต่ส่งคืนผลลัพธ์

Ex03. เขียนโปรแกรมฟังก์ชันคำนวณหาพื้นที่รูปสามเหลี่ยมและพื้นที่รูปวงกลม แล้วส่งผลลัพธ์กลับฟังก์ชัน main()

```
#include  <stdio.h>
float triangle(){
      float  b=10, h=20;
      float area = 0.5 * b * h;
      return area;
}

float circle(){
      float r=7;
      float result = 22.0 / 7.0 * r * r;
      return result;
}

int  main() {
      printf("Area of triangle = %.2f\n", triangle());
      printf("Area of circle = %.2f\n", circle());
      return 0;
}
```

ผลการรันโปรแกรม (Output)

```
Area of triangle = 100.00
Area of circle = 154.00
```

2.4 Arguments and return value: ฟังก์ชันนี้รับข้อมูลอินพุต และส่งคืนผลลัพธ์

Ex04. เขียนโปรแกรมฟังก์ชันหาค่าสุงสุดและค่าต่ำสุด ส่งข้อมูลจากฟังก์ชัน main() ไปทำงานที่ฟังก์ชัน แล้วส่งผลลัพธ์กลับฟังก์ชัน main()

```
#include  <stdio.h>
int max(int x, int y){
      int result;
      result = x > y ? x : y;
      return result;
}

int min(int m, int n){
      int result;
      result = m < n ? m : n;
      return result;
}

int  main() {
      int  a, b;
      scanf("%d%d", &a, &b);
      printf("Max = %d\n", max(a, b));
      printf("Min = %d\n", min(a, b));
      return 0;
}
```

```
if (x > y) return x;
else  return y;
        หรือ
if (x > y) return x;
return y;
```

ผลการรันโปรแกรม (Output)

```
50 45 ↵
Max = 50
Min = 45
```

Ex05. เขียนโปรแกรมฟังก์ชันคำนวณหาพื้นที่รูปสามเหลี่ยมและพื้นที่รูปวงกลม ส่งข้อมูลจากฟังก์ชัน main() ไปทำงานที่ฟังก์ชันที่เขียนขึ้น แล้วส่งผลลัพธ์กลับฟังก์ชัน main()

```
#include  <stdio.h>
float triangle(float b, float h){
      return 0.5 * b * h;
}

float circle(float r){
      return 22.0 / 7.0 * r * r;
}

int  main() {
      float  b, h, r;
      scanf("%f%f", &b, &h);
      printf("Area of triangle = %.2f\n", triangle(b, h));
      scanf("%f", &r);
      printf("Area of circle = %.2f\n", circle(r));
      return 0;
}
```

ผลการรันโปรแกรม (Output)

20 30 ↵
Area of triangle = 300.00
15 ↵
Area of circle = 707.14

Ex06. เขียนโปรแกรมฟังก์ชันแปลงเลขโรมัน (Roman number) เป็นเลขฐานสิบ

เลขโรมัน	เลขฐานสิบ
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

```
#include  <stdio.h>
int revert(char roman){
      switch(roman)
      {
            case 'I'  : case 'i'   : return 1;
            case 'V' : case 'v'   : return 5;
            case 'X' : case 'x'   : return 10;
            case 'L' : case 'l'    : return 50;
            case 'C' : case 'c'   : return 100;
            case 'D' : case 'd'   : return 500;
            case 'M' : case 'm'  : return 1000;
            default : return 0;
      }
}

int  main() {
      char  ch;
      scanf("%c", &ch);
      printf("%d", revert(ch));
      return 0;
}
```

ผลการรันโปรแกรม (Output)

M ↵
1000
i ↵
1
X ↵
10
L ↵
50

Ex07. เขียนโปรแกรมฟังก์ชันโปรแกรมเป่ายิงฉุบ หรือ กรรไกร–ก้อนหิน–กระดาษ (Scissor–Rock–Paper) ระหว่างคอมพิวเตอร์กับผู้เล่น โดยคอมพิวเตอร์จะสุ่มค่าเป็น 0 , 1 หรือ 2 หมายถึง 0–กรรไกร, 1–ก้อนหิน, 2–กระดาษ ตามลำดับ (การสุ่มค่าใช้ฟังก์ชัน rand()) และผู้เล่นจะรับค่าผ่านทางแป้นพิมพ์ เป็น 0 , 1 หรือ 2 จากนั้นจะแสดงผลผลการเล่นเป็น ชนะ (won), แพ้ (lost) หรือ เสมอ (tie/draw)

Input	Output
*** Pao–Ying–Choob *** Scissor(0), Rock(1), Paper(2) : **1**	The computer is scissor. Yor are rock. It is a won.
*** Pao–Ying–Choob *** Scissor(0), Rock(1), Paper(2) : **2**	The computer is paper. Yor are paper too. It is a tie/draw.
*** Pao–Ying–Choob *** Scissor(0), Rock(1), Paper(2) : **0**	The computer is rock. Yor are scissor. It is a lost.
*** Pao–Ying–Choob *** Scissor(0), Rock(1), Paper(2) : **1**	The computer is paper. Yor are rock. It is a lost.

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

int PaoYingChoob(int c, int p){
    char tc[50], tp[50];
    if(c==0) strcpy(tc,"scissor");
    else if(c==1) strcpy(tc,"rock");
    else strcpy(tc,"paper");

    if(p==0) strcpy(tp,"scissor");
    else if(p==1) strcpy(tp,"rock");
    else strcpy(tp,"paper");

    printf("The computer is %s. Yor are %s", tc, tp);
    if(c==0&&p==0 || c==1&&p==1 || c==2&&p==2)
        printf(" too. It is a tie/draw.");
    else if(c==0&&p==1 || c==1&&p==2 || c==2&&p==0)
        printf(". It is a won.");
    else
        printf(". It is a lost.");
}

int main(){
    //0-Scissor, 1–Rock, 2–Paper
    int comp, player;
    srand(time(NULL));      //seed random
    comp=rand()%3;
    printf("   *** Pao-Ying-Choob ***\n");
    printf("Scissor(0), Rock(1), Paper(2) : ");
    scanf("%d", &player);
    PaoYingChoob(comp, player);
    return 0;
}
```

ผลการรันโปรแกรม (Output)

```
   *** Pao-Ying-Choob ***
Scissor(0), Rock(1), Paper(2) : 1 ↵
The computer is scissor. Yor are rock. It is a won.

   *** Pao-Ying-Choob ***
Scissor(0), Rock(1), Paper(2) : 0 ↵
The computer is rock. Yor are scissor. It is a lost.

   *** Pao-Ying-Choob ***
Scissor(0), Rock(1), Paper(2) : 2
The computer is paper. Yor are paper too. It is a tie/draw.
```

Ex08. ตัวแปรแบบโกลบอล (Global Declarations หรือ Global Variable) กับ ตัวแปรแบบโลคอล (Local Declarations หรือ Local Variable)

```
#include <stdio.h>
int a;
void Ex(){
      a = 5;
      printf("%d\n", a);
}

int main(){
      a = 3;
      printf("%d\n", a);
      Ex();
      printf("%d\n", a);
      return 0;
}
```

ผลการรันโปรแกรม (Output)

```
3
5
5
```

```
#include <stdio.h>
int a;
void Ex(){
      int a;
      a = 5;
      printf("%d\n", a);
}

int main(){
      a = 3;
      printf("%d\n", a);
      Ex();
      printf("%d\n", a);
      return 0;
}
```

ผลการรันโปรแกรม (Output)

```
3
5
3
```

```
#include <stdio.h>
int a = 10;
void Ex(){
      int a;
      a = 5;
      printf("%d\n", a);
}

int main(){
      a = 3;
      printf("%d\n", a);
      Ex();
      printf("%d\n", a);
      return 0;
}
```

ผลการรันโปรแกรม (Output)

```
?
?
?
```