

Week 4 เรียนเรื่อง Function

การนิยามฟังก์ชัน

return-value-type function-name(parameter-list)

```
{  
  
    declarations  
  
    statement  
  
    statement  
  
    .  
  
    .  
  
}
```

Function Call – Call by Value

สามารถส่งผ่านค่าข้อมูลจากผู้เรียกไปยังฟังก์ชันที่ถูกเรียกใช้ได้

การส่งผ่านค่าข้อมูล parameter จากผู้เรียกไปยังฟังก์ชันที่ถูกเรียกเรียกว่า

pass by value โดยการส่งผ่านแบบนี้ ฟังก์ชันจะเก็บเพียงค่าข้อมูลที่ส่งมาในฟังก์ชันไว้ในตัวแปรชั่วคราว

Return Values

ฟังก์ชันจะส่งค่ากลับ หรือไม่ก็เป็นฟังก์ชันที่ return type เป็น void

ใช้ void เพื่อบอกว่าไม่มีการส่งค่าใดๆกลับไป แต่หากเป็นฟังก์ชันที่มีการส่งค่ากลับให้ใช้ return เพื่อส่งค่ากลับ เช่น return 5; หรือ return x;

-ค่าที่ได้จากการเรียกใช้ฟังก์ชันสามารถส่งผ่านไปเป็นพารามิเตอร์ของเรียกใช้ฟังก์ชันได้

-ฟังก์ชันสามารถเรียกใช้ตัวเองได้ เรียกว่า Recursion

เมื่อฟังก์ชันใช้ตัวเองจะทำการสำเนาค่าตัวแปรต่างๆภายในฟังก์ชัน

Function Call and Stack Frames

ฟังก์ชันจะ stack frame เพื่อเก็บข้อมูลของตัวแปรพารามิเตอร์และตัวแปรlocalให้อยู่ในฟังก์ชัน เมื่อฟังก์ชันถูกใช้งาน stack frame จะถูกสร้างขึ้นและเมื่อการทำงานเสร็จ stack frame จะถูกทำลาย โดยลำดับของการทำงานจะกลับไปทำคำสั่งที่ return address

return address คือตำแหน่งคำสั่งที่ทำงานหลังจากฟังก์ชันทำงานเสร็จ

ตัวอย่างโค้ดที่เรียน week4

C week4.c

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  int x = 100;
4  int addTen(int x)
5  {
6      x = x + 10;
7      printf("addTen = %d\n", x);
8      return x;
9  }
10 int addTwenty();
11 int main()
12 {
13     int x = 30;
14     int y = addTen(x);
15     printf("addTen main = %d\n", y);
16     int z = addTwenty();
17     printf("addTwenty main = %d\n", z);
18 }
19 int addTwenty()
20 {
21     x += 20;
22     printf("addTwenry = %d\n", x);
23     return x;
24 }
```