Programming Fundamental
Basic C Programming

Cases from Labs

ตัวแปร global , ตัวแปร local

Variables passing in functions

ตัวแปร global

เป็นตัวแปรที่ฟังก์ชันใดก็สามารถเรียกใช้ได้โดยจะประกาศสร้างตัวแปรต่อจาก พรีโปรแซสเซอร์ไดเร็คทีฟ การเปลี่ยนแปลงค่าในฟังก์ชั่นใดๆ จะส่งผลต่อค่าตัวแปร global เสมอ

ตัวแปร local

เป็นตัวแปรที่สามารถเรียกใช้ได้ภายในเฉพาะฟังก์ชันที่ประกาศสร้างตัวแปรนั้นโดยจะประกาศสร้างตัวแปรภายในแต่ละฟังก์ชัน ไม่สามารถเรียกใช้งานได้ในฟังก์ชั่นอื่นๆ

โครงสร้างโปรแกรมภาษาซี

```
#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);
}
```

พรีโปรเซสเซอร์ไดเร็คทีฟ
ฟังก์ชันโพรโทรไทพ์
ตัวแปรชนิดโกบอล
ตัวแปรชนิดโลคอล
คำสั่ง
ฟังก์ชันหลัก
ฟังก์ชันย่อย
ส่วนหัวโปรแกรม
ส่วนตัวโปรแกรม

```
#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
```

ในกรณีที่ชื่อตัวแปรซ้ำกัน
การเรียกใช้ในฟังก์ชั่น จะเรียกใช้ตัวแปร Local เสมอ

ควรใช้ตัวแปร Global เมื่อไหร่ ?
ควรใช้ตัวแปร Local เมื่อไหร่ ?

Example Code

การส่งค่าตัวแปร (pass by value & pass by reference)

การส่งค่าตัวแปรในฟังก์ชั่นมีสองชนิดคือ

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

Pass by Value

- เป็นการส่งค่าที่เก็บอยู่ในตัวแปรเข้าสู่ฟังก์ชัน
- การเปลี่ยนแปลงค่าต่างๆ ของพารามิเตอร์จะไม่เปลี่ยนแปลงค่าของตัวแปรในโปรแกรมหลัก
- ลักษณะของ Pass by Value Function จะไม่ส่งค่า Address (หน้าตัวแปรจะไม่มีเครื่องหมาย *)
- ตัวอย่าง
  - int add(int a, int b)
  - void draw_line(int count)

ให้นักศึกษาสร้างฟังก์ชัน star(int) สำหรับแสดงผลดอกจันจำนวนเท่ากับที่เป็น input แล้วเขียน main() เพื่อรับอินพุทเป็นจำนวนแถว แล้วโปรแกรมแสดงผลเป็นรูปสามเหลี่ยม

Activity

ยกตัวอย่างเช่น

input : 4

output :

****

***

**

*

```
scanf_s (“%d”,&n);
For (i=n;i>0;i--) star(i)
```

ให้นักศึกษาสร้างฟังก์ชัน float area(x1,y1,x2,y2,x3,y3) สำหรับคำนวณพ้นที่ของสามเหลี่ยมที่มีจุดปลายคือ (x1,y1) , (x2,y2) , (x3,y3) โดย x1,x2,x3,y1,y2,y3 เป็น int ทั้งหมด แล้วเขียน main() เพื่อรับอินพุทเป็นตัวเลขทั้ง 6 ตัว

ยกตัวอย่างเช่น

input : 0 0 2 2 2 0

output : 2.00

Activity

```
scanf_s (“%d %d %d %d %d
%d”,&x1,&y1,&x2,&y2,&x3,&y3);
printf_s(“%.2f”,area(x1,y2,x2,y2,x3,y3);
```

Pass by Reference

- เป็นการส่งค่า Address ของตัวแปรเข้าสู่ฟังก์ชัน
- การเปลี่ยนแปลงค่าต่างๆ ของพารามิเตอร์จะส่งผลไปยังตัวแปรในโปรแกรมหลัก
- ลักษณะของ Pass by Reference Function จะส่งค่า Address (หน้าตัวแปรจะมีเครื่องหมาย * หน้าตัวแปรเสมอ)
- ตัวอย่าง
  - int max(int *a, int *b)
  - void increase(int *count)

Pass by Value Function

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

Pass by Reference Function

```
void func2(int *po)
{
 *po = *po +1;
 printf ("In function *po = %d\n",*po);
}
```

Main Function

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

```
Before call function x = 10
In function *pa = 11
After call function x = 11
```

```
int main()
{
  int x;
  x = 10;
  printf ("Before call function x = %d\n",x);
  func(x);
  printf ("After call function x = %d\n",x);
  printf ("\n\n");
  x = 10;
  printf ("Before call function x = %d\n",x);
  func2(&x);
  printf ("After call function x = %d\n",x);

return 0;
}
```

สรุปการเปลี่ยนแปลงค่าตัวแปรในฟังก์ชันหลัก

การทำงานในฟังก์ชัน สามารถเปลี่ยนแปลงค่าตัวแปรในฟังก์ชันหลักได้สองกรณี

1. ตัวแปรที่ส่งค่าเข้าสู่ฟังก์ชัน เป็นตัวแปรแบบ Global Variable
2. การฟังก์ชันทำงานแบบ Pass by Reference

Activity

ให้นักศึกษาสร้างฟังก์ชั่น swap(int *, int*) ที่สามารถสลับค่าที่เก็บอยู่ในตัวแปร 2 ตัว ได้ และเขียนโปรแกรมเพื่อทดสอบการทำงานของฟังก์ชั่น swap

Input:

5 3

Output :

3 5

```
scanf_s (“%d %d”,&a,&b);
swap(&a,&b);
printf_s(“%d %d”,a,b);
```

Example Code