



Jiangxi University of Science and Technology

Ch07 Modularity Using Functions: Part II

Lecture0704 Case Study



Remain Program form last lecture



7.3 Pass by Reference

➤ Program 7.7 **Return a value** by a pointer parameter

```
1.  #include <stdio.h>
2.  int main(){
3.      void newval(float *);
4.      float testval;
5.      printf("\nEnter a number: ");
6.      scanf("%f", &testval);
7.      printf("\nFrom main(): The value in testval is: %5.2f\n", testval);
8.      newval( &testval); /* call the function */
9.      printf("\nFrom main(): The value in testval has been changed to: %5.2f\n", testval);
10.     return 0;
11. }
12. void newval(float *xnum)
13. {
14.     printf("\nFrom newval(): The value pointed to by xnum is: %5.2f\n", *xnum);
15.     *xnum = *xnum + 20.2;
16. }
```

Add 20.2 to the value of the variable pointed to by **xnum**

TEST ME!

7.3 Pass by Reference

➤ Program 7.8 **Returns multiple values** by pointer parameter

```
1.  #include <stdio.h>
2.  int main(){
3.      void calc(float, float, float, float *, float *);
4.      float firstnum, secnum, thirdnum, sum, product;
5.      printf("Enter three numbers: ");
6.      scanf("%f%f%f", &firstnum, &secnum, &thirdnum);
7.      calc(firstnum, secnum, thirdnum, &sum, &product);
8.      printf("\nThe sum of the entered numbers is: %6.2f" , sum );
9.      printf("\nThe product of the entered numbers is: %6.2f\n" , product);
10.     return 0;
11. }

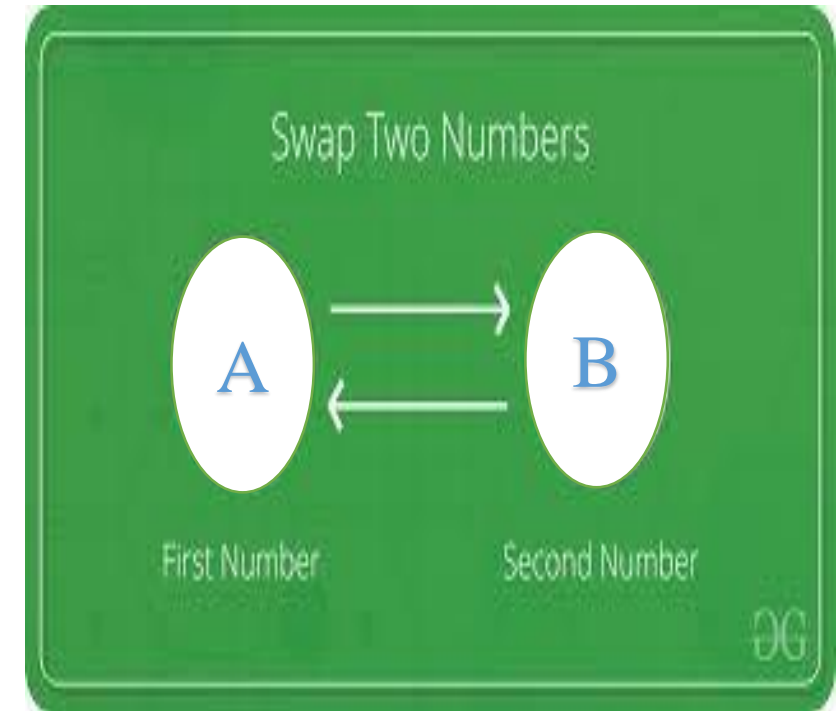
12. void calc(float num1, float num2, float num3, float *sumaddr, float *prodaddr)
13. {
14.     *sumaddr = num1 + num2 + num3;
15.     *prodaddr = num1 * num2 * num3;
16. }
```

TEST ME!

7.4 Case Study: Swapping Values

➤ Swapping Values

- A common programming requirement is the sorting of both numeric values and text, such as names, in either ascending (increasing) or descending (decreasing) order
- Typically accomplished by comparing two values and then **switching values** *if they are not in the correct order*



7.4 Case Study: Swapping Values

➤ Requirements Specification

- Write a C function that exchanges the values in two single-precision variables of its called function
- Thus, if the function has access to two variables of its calling function, the called function should switch the values in these variables

7.4 Case Study: Swapping Values

➤ Analyze the Problem

- **Input** (arguments of the function): **two addresses**, of the two variables whose values are to be exchanged
- **Output**: change the values in the calling function using passed addresses
- **Swapping the values of two variables** is accomplished using the following algorithm:
 - Store the first variable's value in a temporary location
 - Store the second variable's value in the first variable
 - Store the temporary value in the second variable

7.4 Case Study: Swapping Values

➤ Swapping the values of two variables : $A \rightleftharpoons B$
—a temporary variable: **T**

1. **T**=**A**;

2. **A**=**B**;

3. **B**=**T**;



T

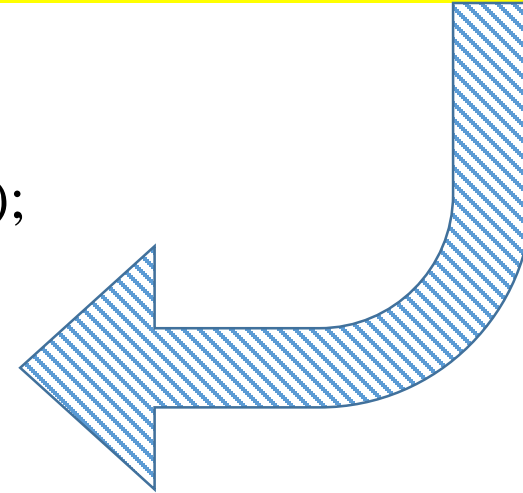


7.4 Case Study: Swapping Values

➤ Program 7.10 Swapping Values

```
1.  #include <stdio.h>
2.  void swap(float *, float *);
3.  int main(){
4.      float firstnum, secnum;
5.      printf("Enter two numbers: ");
6.      scanf("%f %f", &firstnum, &secnum);
7.      printf("\nBefore the call to swap():\n");
8.      printf(" The value in firtsnum is %5.2f\n", firstnum);
9.      printf(" The value in secnum is %5.2f\n", secnum);
10.     swap(&firstnum, &secnum); //call swap()
11.     printf("\nAfter the call to swap():\n");
12.     printf(" The value in firstnum is %5.2f\n", firstnum);
13.     printf(" The value in secnum is %5.2f\n", secnum);
14.     return 0;
15. }
```

```
void swap(float *num1Addr, float *num2Addr){
    float temp;
    temp=*num1Addr; //save firstnum's value
    /* move secnum's value into firstnum */
    *num1Addr=*num2Addr;
    *num2Addr=temp; //change secnum's value
}
```



7.4 Case Study: Swapping Values

➤ Test and Debug the Program 7.10

```
1  #include <stdio.h>
2  void swap(float *, float *); /* function prototype */
3  int main(){
4      float firstnum, secnum;
5      printf("Enter two numbers: ");
6      scanf("%f %f", &firstnum, &secnum);
7      printf("\nBefore the call to swap():\n");
8      printf("  The value in firstnum is %5.2f\n", firstnum);
9      printf("  The value in secnum is %5.2f\n", secnum);
10     swap(&firstnum, &secnum); /* call swap() */
11     printf("\nAfter the call to swap():\n");
12     printf("  The value in firstnum is %5.2f\n", firstnum);
```

Reference



- <https://www.codesdope.com/blog/article/int-main-vs-void-main-vs-int-mainvoid-in-c-c/>

