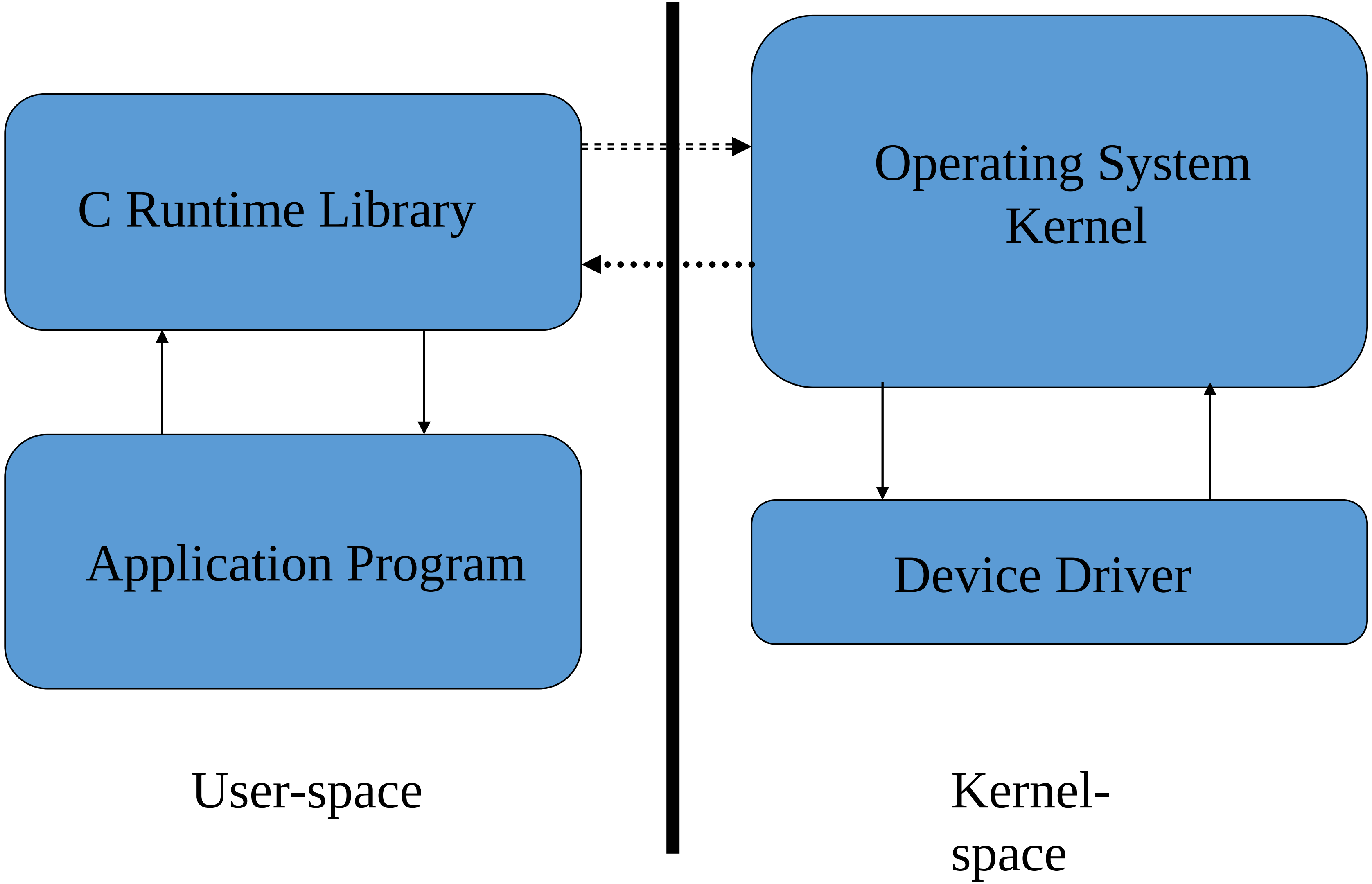
Jiangxi University of Science and Technology

Ch07 Modularity Using Functions: Part II

- Lecture0703 Pass by Reference

How system-calls work



7.3 Pass by Reference

- Pass by value (=Call by value)
  - =**Passing values 传值** from the calling function to the called function
  - A called function receives (copies) values from its calling function, stores the passed values in **its own local parameters**,
  - manipulates these parameters appropriately, and **directly returns, at most, a single value**

7.3 Pass by Reference

- Pass by reference (=Call by reference):
  - =***Passing memory addresses*** 传址 from the calling function to the called function
  - Passing an address is referred to as a function ***pass by reference*** **通过引用传递**, because the called function can ***reference*** or ***access*** the variable using the passed address
  - Also referred to as a call by reference when the term applies only to those parameters whose addresses have been passed

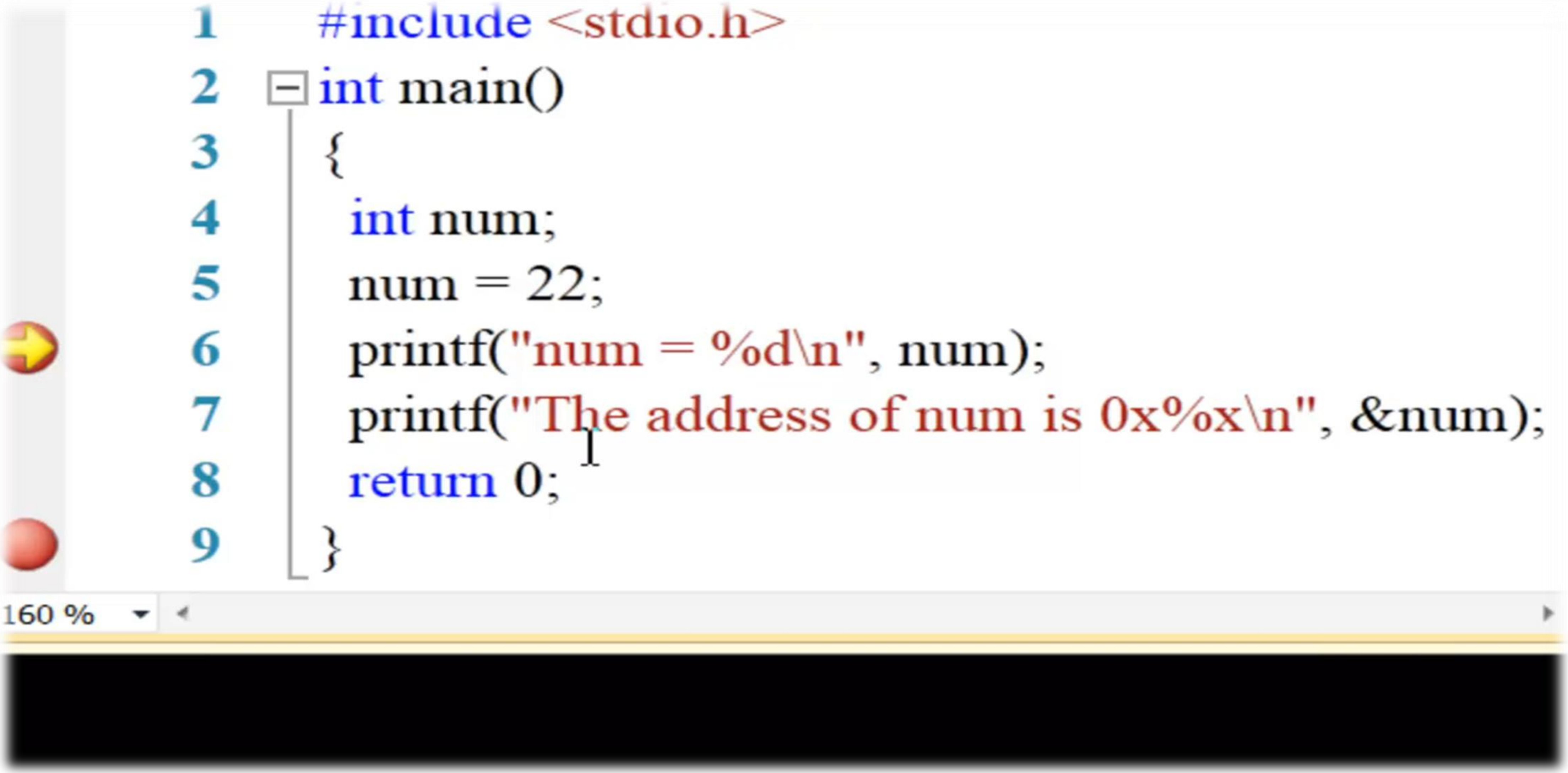
7.3 Pass by Reference

- Program 7.4 Memory Addresses 内存地址

```c
#include <stdio.h>
int main()
{
        int num;
        num = 22;
        printf("num = %d\n", num);
        printf("The address of num is 0x%x\n", &num);
        return 0;
}
```

7.3 Pass by Reference

➢Program 7.4 Memory Addresses

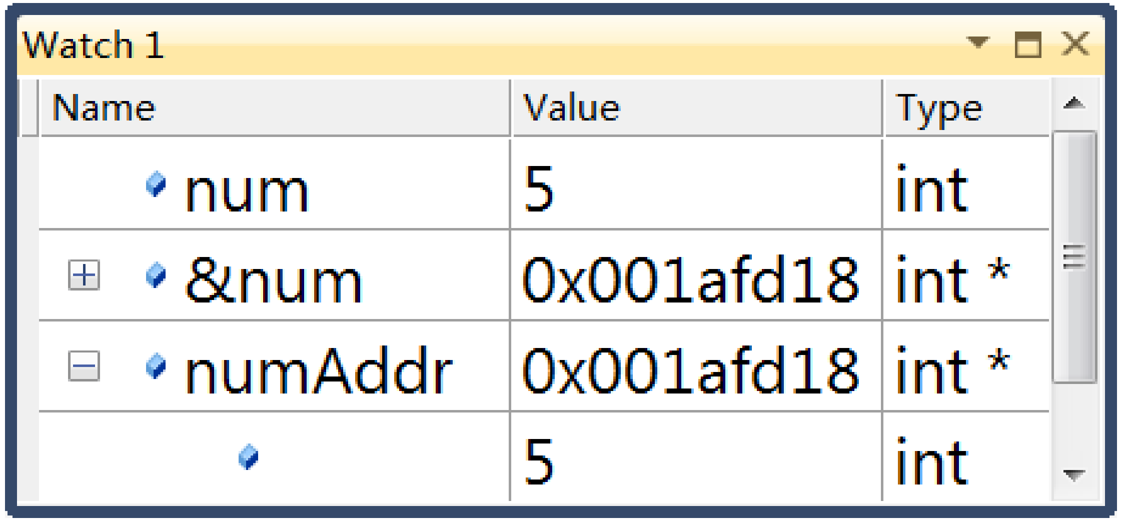


```c
#include <stdio.h>
int main()
{
  int num;
  num = 22;
  printf("num = %d\n", num);
  printf("The address of num is 0x%x\n", &num);
  return 0;
}
```

7.3 Pass by Reference

- Pointer Variable 指针变量

  - **A Pointer stores an memory address !!!**
  - **int** num=5;
  - **int** ***numAddr**;
  - **numAddr**=**&num;**
  - A variable that can store an address is known as **pointer variable** or **pointer**



Watch 1

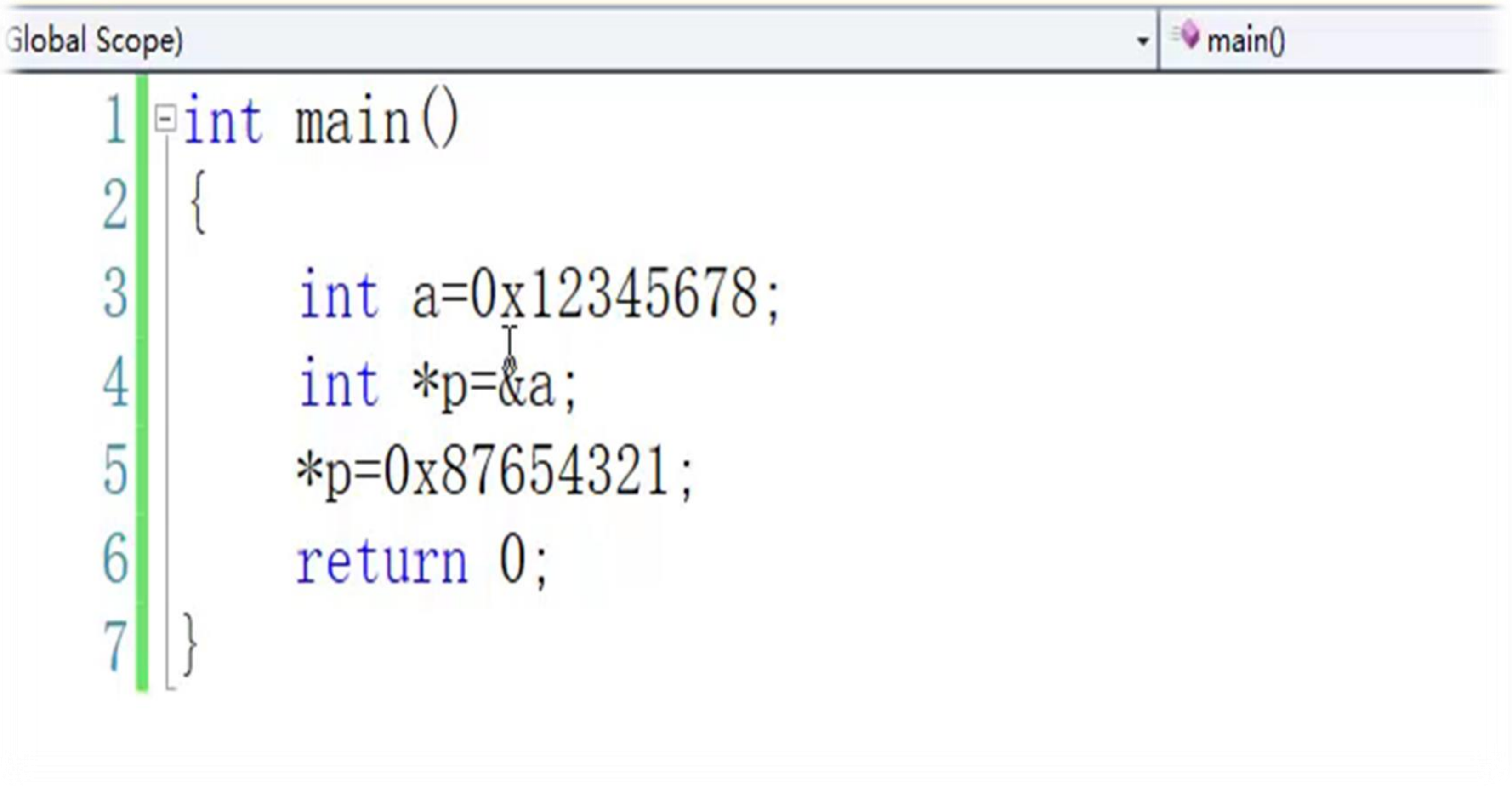
Name	Value	Type
num	5	int
&num	0x001afd18	int *
numAddr	0x001afd18	int *
	5	int

7.3 Pass by Reference

- **Indirection operator**: 间接运算符 * 寻址
  - **int** ***numAddr ;**
  - **float** ***numAddr ;**
  - **means:**
  - numAddr is a pointer
  - ***numAddr** *is the variable whose address is stored in* **numAddr**
  - When using a pointer, the value obtained is always found by first going to the pointer for an address; this is called **indirect addressing** 间接寻址

7.3 Pass by Reference

- * 和&运算符小结
  - **&** 根据变量**取地址**；
  - ***** 根据指针**取变量**；
- 例如



```
int main()
{
    int a=0x12345678;
    int *p=&a;
    *p=0x87654321;
    return 0;
}
```

7.3 Pass by Reference

- ➢ Declaring and Using Pointers

  – In declaring a pointer variable, C requires that we also specify the type of variable that is pointed to

  – **int ***numAddr;

  – This declaration can be read in a number of ways:

    - when the variable pointed to by numAddr is an integer, or when numAddr points to an integer
    - 被numAddr指向的变量是一个整数，或numAddr指向一个整数。

7.3 Pass by Reference

- Program 7.5 Declaring and Using Pointers

```c
#include <stdio.h>
int main(){
        int *milesAddr; /* declare a pointer to an int */
        int miles=22;
        milesAddr = &miles;
printf("The address stored in milesAddr is 0x%08x\n",milesAddr);
printf("The value pointed to by milesAddr is %d\n\n", *milesAddr);
        *milesAddr = 158;
        printf("The value in miles is now %d\n", miles);
        return 0;
}
```

7.3 Pass by Reference

➢Declaring and Using Pointers

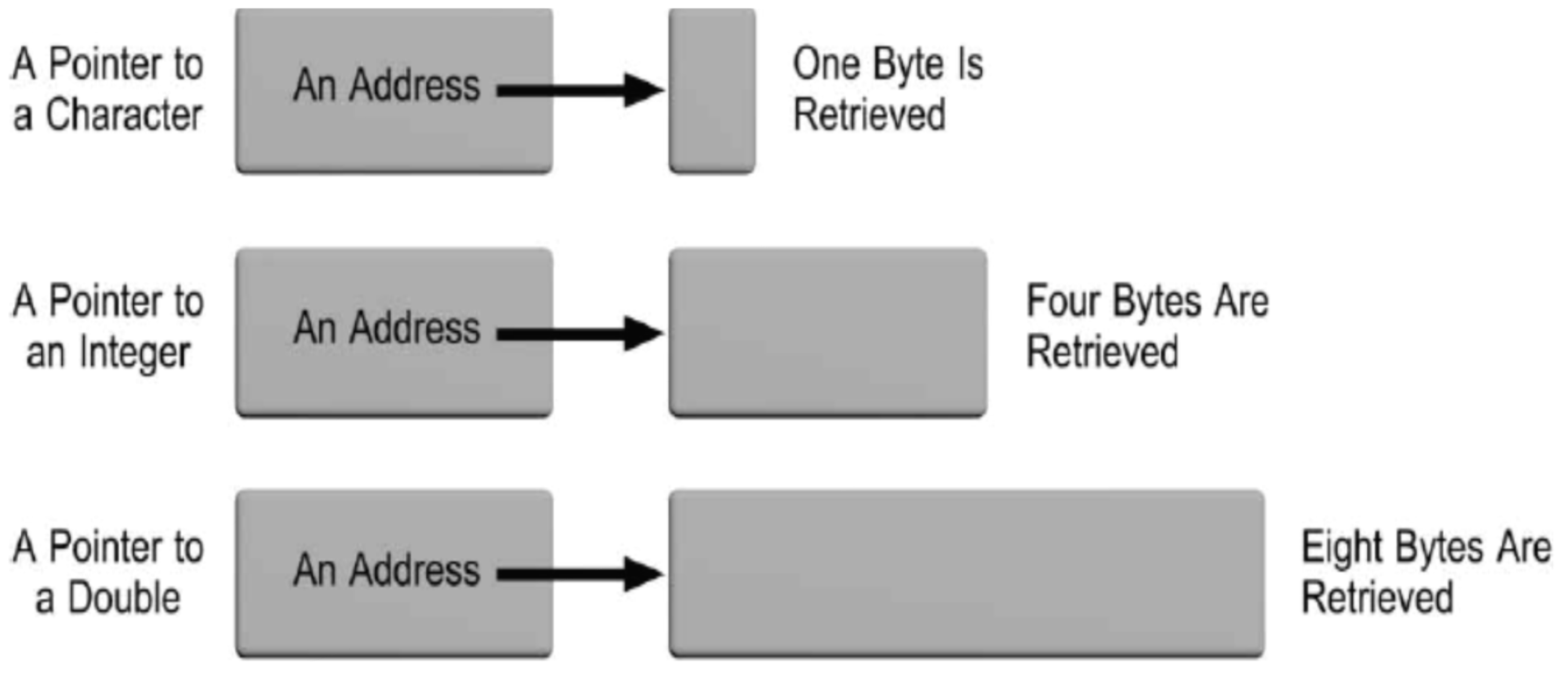
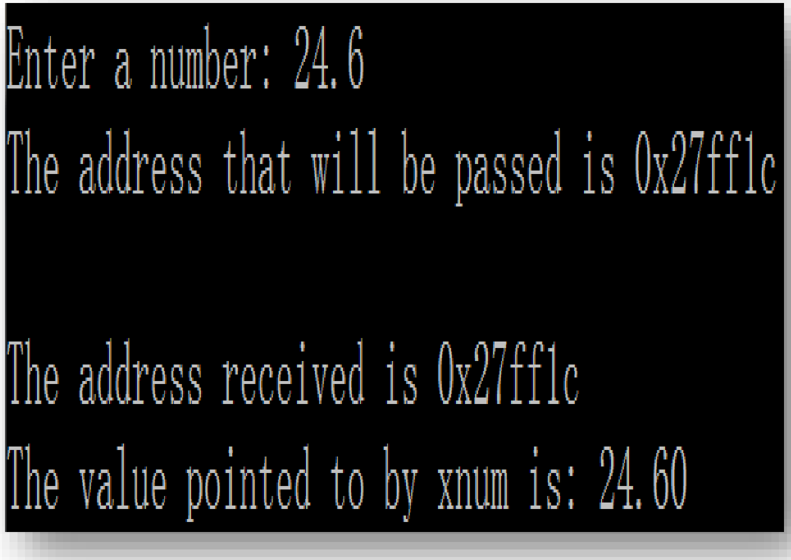


Figure 7.12 Addressing different data types using pointers

7.3 Pass by Reference

Program 7.6 Passing Addresses to a Function

```c
#include <stdio.h>
int main(){
void newval(float *);/*prototype with a pointer parameter */
        float testval;
        printf("\nEnter a number: ");
        scanf("%f", &testval);
        printf("The address that will be passed is 0x%x\n\n", &testval);
        newval(&testval);   /* call the function */
        return 0;
}
void newval(float *xnum)//using a pointer parameter
{
        printf ("The address received is 0x%x\n", xnum);
         printf("The value pointed to by xnum is: %5.2f \n", *xnum );
}
```

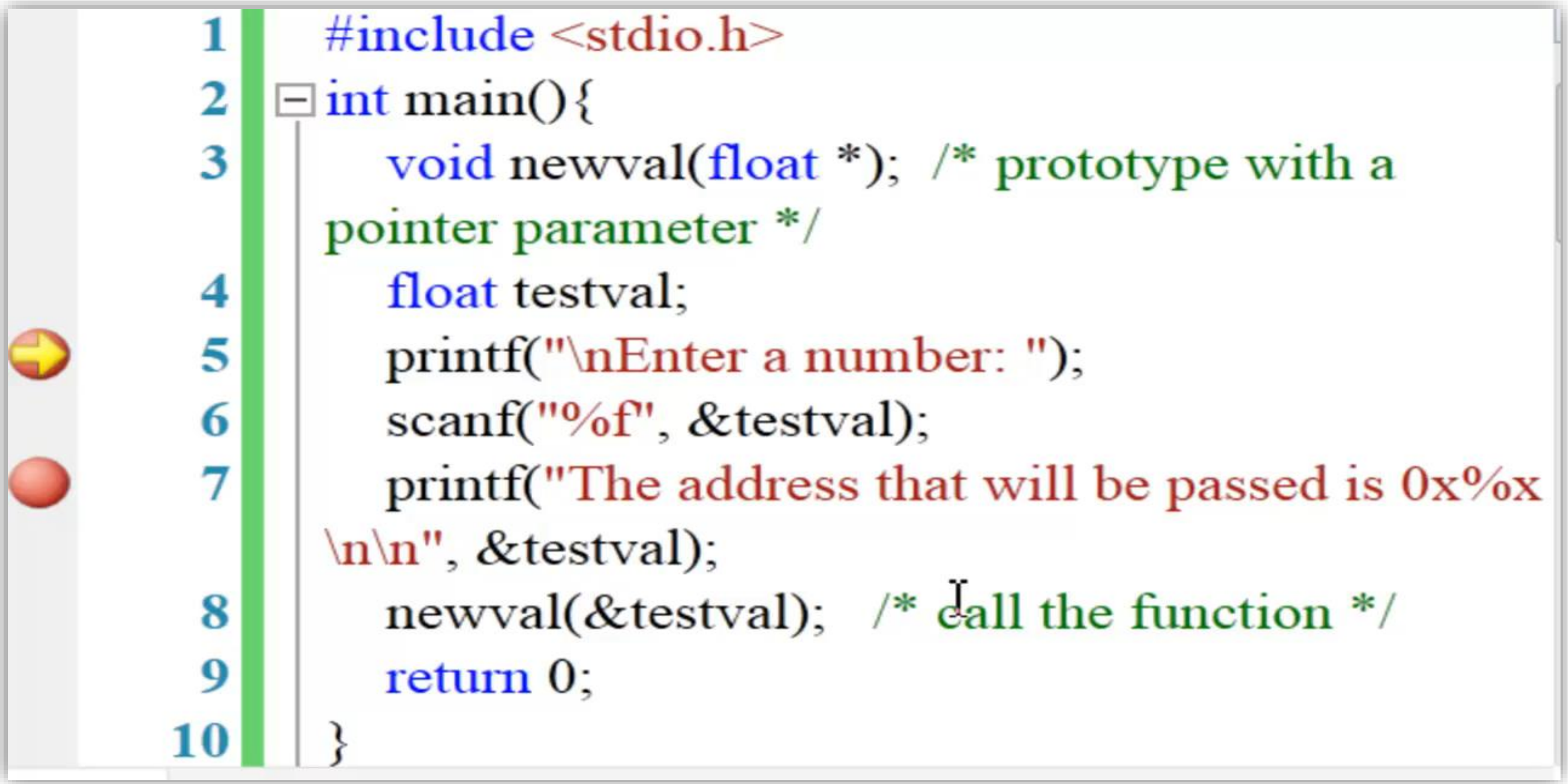


```
Enter a number: 24.6
The address that will be passed is 0x27ff1c

The address received is 0x27ff1c
The value pointed to by xnum is: 24.60
```

7.3 Pass by Reference

➢Program 7.6 Passing Addresses to a Function



```c
#include <stdio.h>
int main(){
    void newval(float *);  /* prototype with a pointer parameter */
    float testval;
    printf("\nEnter a number: ");
    scanf("%f", &testval);
    printf("The address that will be passed is 0x%x \n\n", &testval);
    newval(&testval);   /* call the function */
    return 0;
}
```