

STRUCTURES_II

Passing structure elements to a function in C++

When an element of a structure is passed to a function, you are actually passing the values of that element to the function. Therefore, it is just like passing a simple variable (unless, of course, that element is complex, such as an array of characters). For example, consider the following structure:

```
struct date { int day; int month; int year; } date_1;
```

Individual elements of this structure can be passed as follows:

```
myFunction(date_1.day, date_1.month, date_1.year);
```

The above function-call invokes a function named "myFunction()" by passing values for individual structure elements using the structure variable "date_1".

But remember, if one of the structure elements happens to be an array, it will automatically be passed by reference, as arrays cannot be passed by value.

Consider the following Example:

```
struct date { int day; int month; int year; };
```

```
void printDob(int, int, int); // function prototype
```

```
int main()
```

```
{ date dob;
```

```
cout<<"Enter the Day: ";
```

```
cin>>dob.day;
```

```
cout<<"Enter the Month: ";
```

```
cin>>dob.month;
```

```
cout<<"Enter the Year: ";
```

```
cin>>dob.year;
```

```
printDob(dob.day, dob.month, dob.year); // function call
```

```
cout<<endl; return 0; }
```

```
void printDob(int a, int b, int c) // function definition
```

```
{
```

```
cout<<"\nDate of Birth in DD-MM-YYYY Format = "<<a<<"-"<<b<<"-"<<c; }
```

Structures and Functions

You can pass a structure as an argument of a function in very similar way as you pass any other variable, array or pointer. You would access structure variables in the similar way as you have accessed in the above examples

Ways to pass structure

If the structure itself is an argument, then it is called "call by value". If the reference of the structure is passed as an argument then it is called "call by reference".

- 1- Call by value
- 2- Call by reference

Structures and Functions

1- Call by value

When a structure is passed as argument to a function using call by value method, any change made to the contents of the structure variable inside the function to which it is passed do not affect the structure variable used as an argument.

2- Call by reference

In this method of passing the structures to function, the address of a structure variable/object is passed to the function using address of (&) operator.

So any change made to the contents of structure variable inside the function are reflected back to the calling function.

Consider these examples:

In the following example, we use two functions, f1() has an object argument call by value. Function f2() is call by reference.

Show the output of the following program:

// passing structure to functions

struct sample {int a, b; char c;};

void f1 (sample st); // call by value

void f2 (sample &ss); // call by reference

main()

{ sample s1; s1.a=10; s1.b=20; s1.c='#';

cout<<" output before function "<<s1.a<<" "<<s1.b<<" "<<s1.c<<endl;

cout<<" call by value, so the values of arguments of object not changed"<<endl;

f1(s1);

cout<<" output after call function by value "<<s1.a<<" "<<s1.b<<" "<<s1.c<<endl;

cout<<" call by reference, so the values of arguments of object have been changed"<<endl;

f2(s1);

cout<<" output after call function by reference "<<s1.a<<" "<<s1.b<<" "<<s1.c<<endl; }

void f1(sample st)

{ cout<<" structure in function "<<st.a<<" "<<st.b<<" "<<st.c<<endl;

st.a=7; st.b=9; st.c='m';}

void f2(sample &ss)

{ cout<<" structure in function "<<ss.a<<" "<<ss.b<<" "<<ss.c<<endl;

ss.a=7; ss.b=9; ss.c='\$';}

Output

output before function 10 20 #

structure in function 10 20 #

call by value , so the values of arguments of object not changed

output after first function 10 20 #

structure in function 10 20 #

call by reference, so the values of arguments of object have been changed

output after call function by reference 7 9 \$

Call by reference

```
struct Employee{    char name[50];    int age;    float salary; };
```

```
void readData(Employee &); // Function declaration
```

```
void printData(Employee); // Function declaration
```

```
int main(){    Employee p;    readData(p);    printData(p);    return 0; }
```

```
void readData(Employee &p){
```

```
    cout << "Enter Full name: ";
```

```
    cin.get(p.name, 50);
```

```
    cout << "Enter age: ";
```

```
    cin >> p.age;
```

```
    cout << "Enter salary: ";
```

```
    cin >> p.salary; }
```

```
void printData(Employee p){
```

```
    cout << "\nDisplaying Information." << endl;
```

```
    cout << "Name: " << p.name << endl;
```

```
    cout << "Age: " << p.age << endl;
```

```
    cout << "Salary: " << p.salary; }
```


Output

Enter Full name: Amr Yassen

Enter age: 20

Enter salary: 5000

Displaying Information.

Name: Amr Yassen

Age: 20

Salary: 5000

If calling function readData() by value

Enter Full name: Amr Yassen

Enter age: 30

Enter salary: 10000

Displaying Information.

Name:

Age: 0

Salary: 2.24208e-44

There will be error in the values entered due to call by value

error because no values way assigned to the
variables

Another example of structures &function

- 1- Construct the structure rectangular that contains the dimension of rectangular.
- 2- Write a main function to create two objects of the above structure and use a function to read the data of the two objects.
- 3- Use another function to print the data of the two objects.

// If function call by value the values will not be transferred correctly to the main, it must be call by reference

```
struct rectangular {float l, w;};
```

```
void read_R(rectangular &R) // must be call by reference  
{ cout<<"Enter data of rectangular"<<endl;  
  cin>>R.l >> R.w;          }
```

```
void print_R (rectangular R) // can be call by value or call by reference  
{cout<<"length= "<<R.l<<" width= "<<R.w<<endl; }
```

```
int main( )  
{ rectangular R1, R2;
```

```
  read_R(R1);  
  read_R(R2);  
  cout<<" print data of rectangular R1 "<<endl<<endl;  
    print_R(R1);  
  cout<<" print data of rectangular R2 "<<endl<<endl;  
    print_R(R2);          return 0;}
```

هالام

struct mustbe
defined before
functions that iuse
it

Output if the dimension as follows:

Enter data of rectangular

10.55 6.7

Enter data of rectangular

12.8 4.8

print data of rectangular R1

length= 10.55 width= 6.7

print data of rectangular R2

length= 12.8 width= 4.8

Example of structure to define information of books

1- Define a structure of information of book as:

- *Title as an array of characters*
- *Author name as an array of characters*
- *Subject of book as an array of characters*
- *number of book as integer*

2- *Construct two objects of the structure and assign their input data.*

Solution

struct Books

```
{ char title[50];  
  char author[50];  
  char subject[100];  
  int book_id; };
```

int main()

```
{ struct Books Book1;    // Declare Book1 of type Book  
  struct Books Book2;    // Declare Book2 of type Book  
  strcpy( Book1.title, "Learn C++ Programming");  
  strcpy( Book1.author, "Chand Miyan");  
  strcpy( Book1.subject, "C++ Programming");  
  Book1.book_id = 6407;
```

// book 2 specification

```
strcpy( Book2.title, "Telecom Billing");  
strcpy( Book2.author, "Yakit Singha");  
strcpy( Book2.subject, "Telecom");  
Book2.book_id = 64700;
```

From the above example of structure Books

Write a function that prints the information of any book, then use the function in the main() function to print the data of the two books

```
void printBook( Books book ) // can be call by value  
{  
    cout << "Book title : " << book.title <<endl;  
    cout << "Book author : " << book.author <<endl;  
    cout << "Book subject : " << book.subject <<endl;  
    cout << "Book id : " << book.book_id <<endl; }
```

In main

```
// Print Book1 info  
printBook( Book1 );  
  
// Print Book2 info  
printBook( Book2 );
```

Output

Book title : Learn C++ Programming

Book author : Chand Miyan

Book subject : C++ Programming

Book id : 6407

Book title : Telecom Billing

Book author : Yakrit Singha

Book subject : Telecom

Book id : 64700

Example: function returns a structure

Create a structure that contains the name and price of certain product

Write a main program to create two objects of the structure and use two functions, function `addname ()` to enter the data of the product and returns its object. Function `list ()` to print the data of the objects

// passing structures to functions and a function returning a structure

// -----structure part---

struct product

{ string name; float price;};

// function prototype of type struct

struct product addname(); *//the word struct can be removed*

// another function prototype

void list_func(product);

هالام

```
// -----main program-----  
int main()  
{ // declare 2 structure variables  
    product p1,p2;  
    // functions call for user input...  
    cout<<"Enter first product "<<endl;  
    p1 = addname();  
    cout<<"enter second product "<<endl;  
    p2 = addname();  
    cout<<"\nproduct for sale\n";  
    // function call for data display...  
    list_func(p1);  
    list_func(p2);  
    cout<<endl;    return 0;}
```

قالام

//-----function-----

// this functions returns a structure

struct product addname() //It can be written directly product addname()

```
{ // declare a structure variable
  product pnew;
  cout<<"\nEnter name of product: ";
  getline(cin,pnew.name);
  cout<<"Enter price :LE ";
  cin>>pnew.price;      return pnew;}
```

// structure passed from main()

void list_func(product list)

```
{  cout<<"\nproduct name: "<<list.name;
   cout<<"\nproduct price: LE "<<list.price; }
```

Solution:

Output

Enter first product

Enter name of product: kitchen machine

Enter price :LE 900

Enter second product

Enter name of product: kettle

Enter price :LE 300

product for sale

product name: kitchen machine

product price: LE 900

product name: kettle

product price: LE 300

Assignment

1- Complete the following program that uses the following declared structure student which contains the student number, name, his scores in m subjects. Then the program creates n objects of the structure student, read data for n students, get average score using `ave()` function and search for certain student by his name using the search function. The main program prints his name and scores.

```
Struct student { int id; char name[30]; float score[10];};
```

```

#.....
#.....
#.....
..... search (..... )
{   for(int i=0;i<n; i++)
.....
.....}
.....average(.....)
{ float av=0; .....
.....
.....}

main( )   {int i, j, k, n,m; student.....;   char target[30];cin>> n>>m;
    for (i=.....)
    {cin>>.....; gets(.....)
        for(j=.....)          cin>>.....;}
    cout<<" enter name  ";
    gets(target); k = search(.....);
    if(k< 0) cout<<.....;
    else{ cout<<" the student was found at "<<.....<<endl;
.....
.....
.....}

```