

Chapter 3 : Structures and Files in C

We learned in the previous chapter how to store variables that have the same data type as elements in one block called an array. Currently, the questions arise are: how about elements with different data type ? What is the form of the item that combines these elements? How we can deal with this item? In this chapter, we will try to answer these questions in detail. Besides that we will discuss the files in C to touch their advantages in programming.

Part I : Structure

Definition :

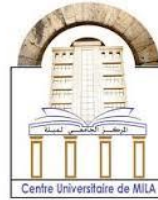
A structure is a collection of elements that carry inhomogeneous data types. The element in a structure is called a member. Likewise, an array is a method to rejoin many variables in a single entity. The difference between structure and array is that the elements of an array have the same type unlike structure members. Also, there is a difference in access mode. In an array, an index is required to access an element, but access to a member in the structure is done by referring to them by their name.

Creation of a structure :

To create a structure, the key **struct** is used in a syntax as follows

```
struct name_of_structure {  
    data-type member1 ;  
    data-type member2 ;  
    ..... ;  
    .....  
};
```

This can be considered as a simple declaration, and unlike usual, nothing is stored in memory. For a memory allocation to this structure, a declaration of a variable to this structure is needed.



Computer Science 2

Example 1:

For example the structure variable here is person1

```
#include<stdio.h>
struct user{
char name[20];
int password;
};

int main()
{
struct user person1 ;
.....
return 0 ;
}
```

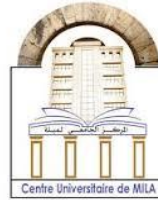
Access to structure members :

Access to the second member in the structure from the previous example is done like this **person1.password**.

Example 2 : (Example of a structure)

Program	Output
<pre>#include<stdio.h> #include<string.h> struct user{ char name[20]; long password; }; int main() { struct user person1 ; strcpy(person1.name,"Moussa"); person1.password=112233; printf("Name: %s\n", person1.name); printf("Password:%d", person1.password); return 0 ;}</pre>	<pre>Name: Moussa Password:112233</pre>

To assign values to a string in a structure one can use this `person1.name="Moussa"` but this is not work here. The right way is to use the function `strcpy(,)` from the standard library `string.h`.



Computer Science 2

Example 3 :

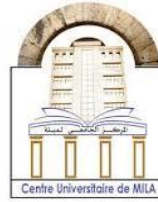
This example shows a simple way to assign values to a structure members

Program	Output
<pre>#include<stdio.h> struct user{ char name[30] ; int age ; char blod_type[5] ; long password; }; int main() { struct user person1={"Ahmed",28,"A+",122343} ; printf("Name: %s\n", person1.name); printf("Age:%d\n", person1.age); printf("Blod type:%s\n", person1.blod_type); printf("Password:%d", person1.password); return 0 ; }</pre>	<pre>Name: Ahmed Age:28 Blod type:A+ Password:122343</pre>

One of the advantages of a structure is the ability to create multiple variables, and this can be shown in this example.

Example 4 :

Program	Output
<pre>#include<stdio.h> struct user{ char name[30] ; int age ; char blod_type[5] ; long password; }; int main(){ struct user person1={"Ahmed",28,"A+",122343}; struct user person2={"Kamel",30,"B-",316235}; printf("Name: %s\n", person1.name); printf("Age:%d\n", person1.age); printf("Blod type:%s\n", person1.blod_type); printf("Password:%d", person1.password); printf("\n-----\n"); printf("Name: %s\n", person2.name); printf("Age:%d\n", person2.age); printf("Blod type:%s\n", person2.blod_type); printf("Password:%d", person2.password); return 0 ; }</pre>	<pre>Name: Ahmed Age:28 Blod type:A+ Password:122343 ----- Name: Kamel Age:30 Blod type:B- Password:316235</pre>



Computer Science 2

The program mentioned above seems to be unsuitable and requires a loop.

Example 5 : for loop in a structure

<i>Program</i>	<i>Output</i>
<pre>#include<stdio.h> struct user{ char name[30] ; int age ; char blod_type[3] ; long password; }; int main() { Struct user person[2]={{"Ahmed",28,"A+",122343}, {"Kamel",30,"B-",316235}}; int i; for(i=0;i<2;i++) { printf("Name: %s\n", person[i].name); printf("Age:%d\n", person[i].age); printf("Blod type:%s\n", person[i].blod_type); printf("Password:%d\n---\n", person[i].password); } return 0 ; }</pre>	<pre>Name: Ahmed Age:28 Blod type:A+ Password:122343 --- Name: Kamel Age:30 Blod type:B- Password:316235 ---</pre>