|  |  |  |
| --- | --- | --- |
| **CL1002 Programming Fundamentals** | **LAB 10**  **Structures and Nested Structure** | |
|  | |  |

**Learning Objectives**

1. Structures
2. Nested Structures
3. **Structures**

Structures are derived data types—they’re constructed using objects of other types. Normally, we use structure to store the record or the details of any item or entity. Structure members can be variables of the primitive data types (e.g., int, float, etc.), or aggregates, such as arrays and other structures.

* Keyword struct introduces a structure definition
* The identifier Chocolate is the structure tag, which names the structure definition and is used with struct to declare variables of the structure type—e.g., struct Chocolate kitkat, Mars, Jubilee.
* Variables declared within the braces of the structure definition are the structure’s members.
* Members of the same structure type must have unique names, but two different structure types may contain members of the same name without conflict.
  1. **Declaration of Struct**

struct Choclate{

char Name[20];

float Weight;

int Calories;

float Price;

char ExpiryDate[10];

};

* 1. **Declaration & Initialization of Struct type Variables**

You can declare the variables before the semi-colon(;) or using a proper declaration syntax like other variable’s in main();

struct Chocolate{

char Name[20];

float Weight;

int Calories;

float Price;

char ExpiryDate[10];

}var1, var2,var3;

Int main()

{

Struct Chocolate Kitkat, Mars, Jubliee, mychocolate[3];

// OR

struct Chocolate myChocolate;

gets(myChocolate.Name);

myChocolate.Weight= 20;

myChocolate.Calories= 500;

myChocolate.Price= 100;

strcpy(myChocolate.ExpiryDate,"01-Feb-2021");

// OR

struct Chocolate Jubliee = {"Jubilee",20.50,500,100,"01-Feb-2021"};

}

* 1. **Declaration & Initialization of Struct type Array**

Int main()

{

// Array of Struct

struct Chocolate myFavChocolates[3]; // It is an array of struct

int i = 0;

while(i<3)

{

gets(myChocolate[i].Name);

scanf("%f",&myChoclate.Weight);

scanf("%d",&myChoclate.Calories);

scanf("%f",&myChoclate.Price);

gets(myChoclate[i].ExpiryDate);

++i;

}

// TO print this array of Struct

i = 0;

while(i<3)

{

puts(myChoclate[i].Name);

printf("%f",myChoclate.Weight);

printf("%d",myChoclate.Calories);

printf("%f",myChoclate.Price);

puts(myChoclate[i].ExpiryDate);

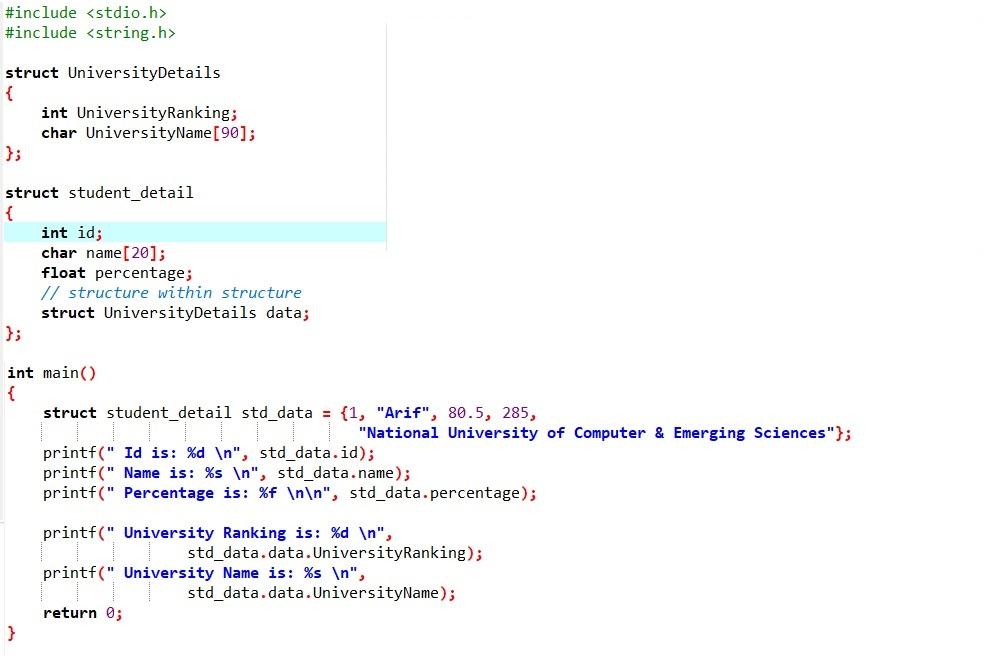
++i;

}

}

**2.0 Nested Structures**

Nested structure in C is nothing but structure within structure. One structure can be declared inside another structure as we declare structure members inside a structure. The structure variables can be a normal structure variable ,array or a pointer variable to access the data. You can learn the concepts below in this section.



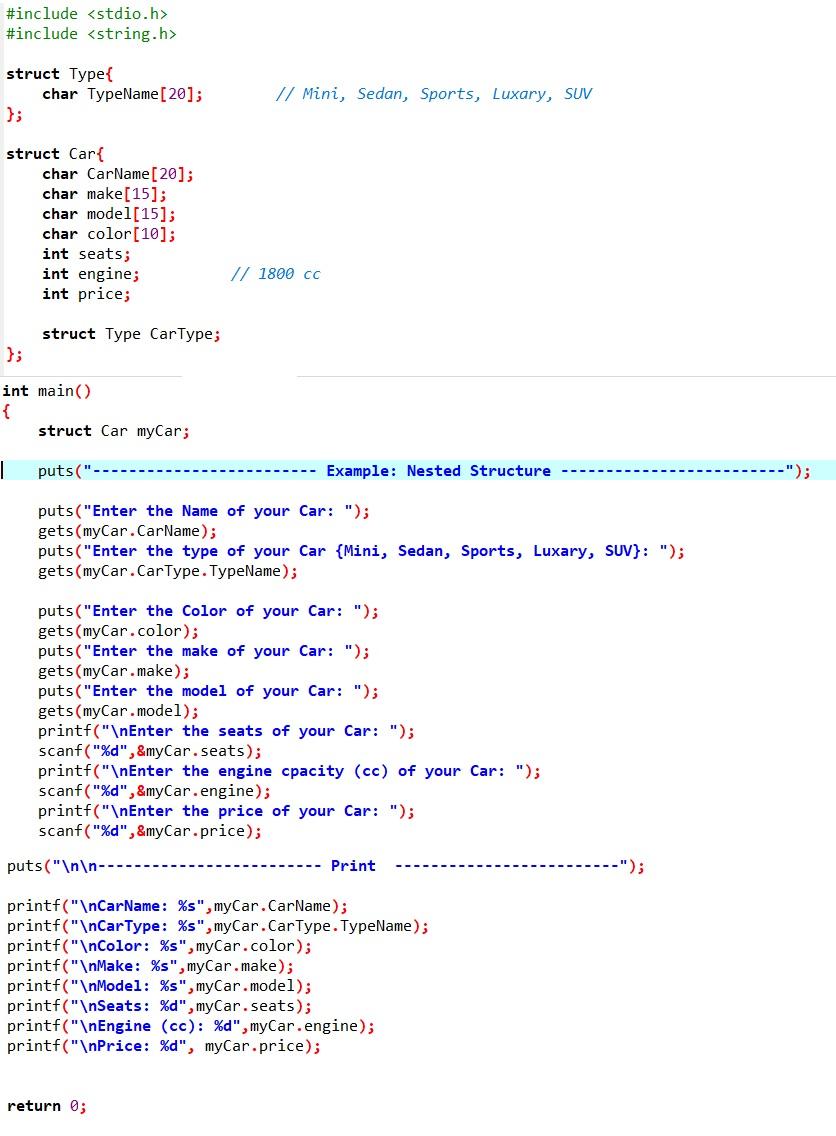
**OUTPUT:**

**Text

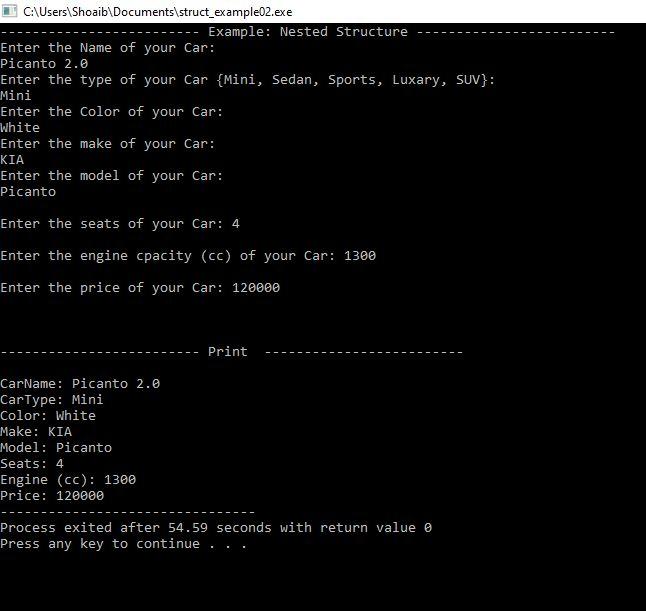
Description automatically generated**

Another example of Nested Structure:

**Sample Code:**



**OUTPUT:**

****