17

STRUCTURE, TYPEDEF & ENUMERATED DATA TYPE

- **STRUCTURE**: It is an user-defined data type available in C/C++. It allows a user to combine data items of different data types under a single name.

Structures are different from arrays because arrays only hold data of similar data-types. C++ structure, on the other hand,can store data of multiple data-types.

Each element in the structure is called a member.

- **SYNTAX:** A structure is defined with the **struct** keyword. A structure is a possible collection of primary data types and other structures.
  - The structure_name holds the name we want to give to our structure.
  - data_type variable is the C++ variables of different data types like int, char, float, etc.

```
#include <iostream>
using namespace std;

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

int main()
{
    Person p1;

    cout << "Enter Full name: ";
    cin.get(p1.name, 50);
  cout << "Enter age: ";
  cin >> p1.age;

    cout << "Enter salary: ";
    cin >> p1.salary;

    cout << "\n Displaying
Information." << endl;
    cout << "Name: " << p1.name <<
endl;
    cout <<"Age: " << p1.age << endl;
    cout << "Salary: " << p1.salary;

    return 0;
}
```

Output:

```
Enter Full name: Sourav Nanda
Enter age: 27
Enter salary: 45000

Displaying Information.
Name:  Sourav Nanda
Age: 27
Salary: 45000
```

From the above example: P1 is the structure variable
To access the

- **VARIABLE OF THE STRUCTURE:** P1 is the structure variable and it allocates 56 bytes of memory
- **ACCESSING OF DATA MEMBERS:**

structure variable. member name such as P1.name, P1.age, P1.salary

- **STRUCTURE WITH IN STRUCTURE/ NESTED STRUCTURE:**. One structure can be declared inside other structure as we declare structure members inside a structure. The structure variables can be a normal structure variable to access the data.

```
#include <iostream.h>
Using namespace std;
Struct date_of_birth{
int dd,mm,yy};
Struct student{
char name[30];
int rollNumber;
date_of_birth dob;
};
int main(){
student S;
cout<<"Enter name :";
cin.getline(S.name);
cout<<"Enter rollnumber :"
cin>>S.rollNumber;
cout<<"Enter date of birth (dd mm yy)
:";
cout<<"Name:"<<S.name<<",Roll
Number:"<<S.rollNumber<<endl;
cout<<"Date                        of
birth:"<<S.dob.dd<<"/"<<S.dob.mm<<
"/"<<S.dob.yy<<endl;
return 0;
```

Output:

```
Enter name : Pooja

Enter rollnumber :001

Enter date of birth : 15-01-2007

Name: Pooja Roll Number: 001 Date of
Birth: 15-01-2007
```

- **TYPEDEF:** The typedef in C/C++ is a keyword used to assign alternative names to the existing datatypes. It is used to define a new name for an existing data type
  Syntax: typedef existing datatype new datatype;
  For example: typedef float real;

- **ENUMERATED DATA TYPE:**
  - It is a user defined data type.
  - It works if finite list is known. The list cannot be input by the user or output on the screen.

  For example:

enum months { jan, feb, mar, apr, may};
enum days { sun, mon, tue, wed, thu };

The first name was given the integer value 0, the second value 1 and so on.

- jan = 0, feb = 1, mar = 2, apr = 3, may = 4
- The ordering can be altered by using an equal sign and value.
- enum months { jan = 1, feb, mar, apr, may };
  Here jan = 1, feb = 2, mar = 3, apr = 4, may = 5

```
    #include <iostream.h>
    Using namespace std;

enum months { jan, feb, mar, apr, may };
void main ( ) {

 months m1, m2; m1 = jan; m2 = apr;

 int diff = m2 - m1;

 cout << "Months between" << diff << "\n";
if (m1 > m2)

cout << "m2 comes before m1";

 }
```

Output:

```
Months between 3
```

CHECK YOURSELF

1. The data elements in the structure are also known as ?
A) objects C) data
B) members D) objects & data

2. What will be used when terminating a structure?
A) : C) ;
B) } D) ;;

3. Which of the following keywords is used to define an alternate name for an already existing data type?
A) Default
B) Volatile
C) typedef
D) Static

4. What will happen when the structure is declared?
A) It will not allocate any memory
B) It will allocate the memory
C) It will be declared and initialized
D) It will be declared

5. typedef declaration:

A. Does not create a new type

B. It merely adds a new name for some existing type.

C. Both a & b

D. None of the mentioned

STRETCH YOURSELF

1. Define structure. Explain where and why to use structure?
2. What is nested structure? Give an example.
3. Differentiate between data type and typedef?

ANSWERS

Answers to Check Yourself:

1. B
2. C
3. C
4. A
5. C