12

Structure, Typedef & Enumerated Data Type

12.1 Introduction

Sometimes it is required to store information, which can be of same or different types. For this purpose, structure statement is used. This lesson contains the three important statements of C++ language.

12.2 Objectives

After going through this lesson, you would be able to:

- define variable, initialize and access members of the structure
- explain the concept of nested structure
- explain typedef statement in a program
- define enum statement and use it

12.3 Structure

A structure is a collection of simple variables which can be of same or different types. The data items in a structure are called the members of the structure. Consider the following example:

```
        # include < iostream.h >
```

```
struct student
{
    char name [20];
    int marks;
};
    void main ( )
{
    student S1, S3;
    cin >> S1. name >> S1. marks;
    cout << S1. name << S1. marks;
    student S2 = {"ANKIT", 32};
    cout << S2. name << S2. marks;
    S3 = S2;
    cout << S3. name << S3. marks;
    }
```

12.3.1 Variable of the structure

S1 and S2 are called variable of the structure student. They are given enough space in memory to hold the members of the structure (22 bytes each for above program).

12.3.2 Accessing of data members

The accessing of data members is done by using the following format:

```
structure variable. member name
```

12.3.3 Initialization of structure variable

Initialization is done at the time of declaration of a variable. For example:

```
Student S2 = {"ANKITA", 32};
```

12.3.4 Structure variable in Assignment Statements

The statement

```
S3 = S2;
```

assigns the value of each member of S2 to the corresponding member of S3. Since a large structure can have many members, such an assignment statement requires the compiler to do a considerable amount of work.

Note that one structure variable can be assigned to another only when they are of the same structure type, otherwise compiler will give an error.

12.3.5 Structure within Structure

This is called nesting of structure. Consider the following program in which date field is a structure.

```
# include < iostream.h >
# include < stdio.h >
struct today
{
int month;
int date;
int year;
};
struct biodata
{
char name [ 20 ];
today date_of_birth;
char add [ 30 ];
};
void main ( )
{
biodata b;
cout << “Enter name”;
gets (b.name);
cout << “Enter birth month”;
cin >> b.date_of _birth. month;
cout << “Enter birth date”;
cin >> b.date_of_birth.date;
cout << “Enter birth year”;
cin >> b.date_of _birth.year;
cout << “Enter address”;

gets (b.add);

cout << “Name” << b.name << “\n”;
cout << “Month” <<b.date_of_birth.month << “\n”;
cout << “Date” << b. date _of_birth. date << “\n”;
cout << “Year” << b. date_of_birth.year << “\n”;
cout << “Address” << b.add << “\n”;
}
```

12.3.6 Accessing nested structure members

As the structure is nested inside another, we must apply the dot operator twice to access the structure members:

cin >> b. date_of_birth. month;

12.3.7 Initializing Nested Structure

It is initialized at the time of declaration of structure variable. For example:

biodata bd = {“MOHIT”, {3, 21, 1991}, “Rohini”};

The inner braces are required just to separate the values, although it is ignored by the compiler.

In-Text Questions 12.1

1. What is a structure?
2. How is structure different from array?
3. The variable of the structure is given space in the memory. True or False?
4. What is the syntax of accessing data members of the structure?
5. The initialization of a structure is done at the time of assigning data type. True or False?
6. What is nesting of structure?
7. Write a structure specification that includes three variables all of type int called hrs, mins and secs. Call this structure time.
8. Write a statement that sets the hrs member of the time 2 structure variable equal to 11.
9. If you have three variables defined to be a type struct time and this structure contains three int members, how many bytes of memory do the variables use together?

10. Write a C++ nested structure for the record given below. Use appropriate data type.

Name	Class	Roll	Address
			House No
			Street
			City
			State
			PIN Code

Declare a variable of the structure. Assign the value of each variable in a nested structure.

11. Write statements in the following program segment that will assign values to all the fields of a structure.

```
struct date
{
int dd, mm, yy;
};
struct donor
{
char N [ 20 ];
date dob;
};
void main ( )
{
donor person;
-----------------
-----------------
}
```

12. What will be the output of the following programs.

```
    (a)   # include < iostream.h >
          void main ( )
          {
          struct rollno
          {
          int roll;
          };
          rollno r;
          r. roll = 100;
          cout << “roll = ” << r. roll;
    }
```

```
(b)   # include < iostream.h >
      void main ( )
      {
      struct student
      {
      int roll;
      int age;
      };
      student S1 = { 100, 15 };
      student S2 = S1;
      if ( S1 = = S2 )
      cout << “The structures are equal”;
      else
      cout << “The structures are not equal”;
      }
```

12.4 Typedef

It is used to define new data type for an existing data type. It provides an alternative name for standard data type. It is used for self documenting the code by allowing descriptive names. (for beltes understanding) for the standard data type.

The general format is:

typedef existing datatype new datatype;

For example:

typedef float real;

Now, in a program one can use datatype real instead of float.

Therefore, the following statement is valid:

real amount;

In-Text Questions 12.2

1. What is the function of typedef statement?
2. Why typedef statement is used in a program?
3. What is the meaning of the following statement?

   typedef int integer:

12.5 Enumerated Data Type

It has the following features:

- It is user defined.
- It works if you know in advance a finite list of values that a data type can take.
- 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 };

enum toys { cycle, bicycle, scooter };
```

The enum specifier defines the set of all names that will be permissible values of the type called members which are stored internally as integer constant. The first name was given the integer value 0, the second value 1 and so on.

For example:

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

The value of the next element in the list is previous value plus one.

For example:

```
# include < iostream.h >
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”;
}
```

The output of the above program is

```
Months between 3
```

because the members of the enumerated series is stored as integer constant in the memory.

In-Text Questions 12.3

1. An enumerated type is considered as an integer type internally. True of False?

2. What is the meaning of following statement?

   enum days { sun, mon, tue };

3. Can the enum data input by the user?

4. (a) Write a statement that declares an enumerated data type called months with the values

   jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec

   (b) Define two variables m1 and m2 and assign them the values jan and mar respectively.

5. The first three members of an enum type are normally represented by the values ...., ... & ....

6. Write a statement that declares an enumerated data type called colour with the values pink, red, green, and blue. Give these four names the integer values 33, 45, 58 and 78.

12.6 What you have learnt

In this lesson, you learnt about three statements: struct, typedef and enum. Now you should be able to use these statements in C++ program.

12.7 Terminal Questions

1. Write a C++ nested structure for the record given below. Use appropriate data type.

Name	Roll No.	Date of Birth
		Day
		Month
		Year

   Assign value to all the members of the structure.

2. What will be the output of the following program?

```
# include < iostream.h >
enum days { sun, mon, tue, wed, thu, fri, sat };
void main ( )
{
days today;
```

```
        for (today = sun; today < = sat; today ++ )
        cout << "\n today is "  << today;
        }
```

3. State the reason why

   enum boolean {false, true};
   is better than
   enum boolean {true, false};

4. What is the advantage of typedef statement?

5. A phone number, such as 786-6370, can be thought of having two parts; the exchange code and the number. Write a program in C++ that uses a structure phone to store these two parts. Call the structure phone. Create two structure variable of type phone. Initialize one and have the user input a number for the other one. Then display both the numbers.

12.8 Feedback to In-Text Questions

In-text Questions 12.1

1. Structure is a collection of simple variables which can be of same or different types.

2. An array can hold variables of same data types whereas structure can hold variables of same or different data types.

3. True

4. structure variable. member

5. True

6. The structure within a structure is called nesting of structure.

7. struct time

```
{
int hrs;
int mins;
int secs;
};
```

8.

```
time time2;

time2. hrs = 11;
```

9. 18

10.

```
struct add

{
char houseno [ 5 ];
char street [ 10 ];
char city [ 10 ];
char state [ 10 ];
char zip [ 10 ];
};
struct biodata
{
char name [ 20 ];
char class [ 5 ];
into roll;
add address [ 30 ];
};
biodata data;
cin > > data. name;
cin > > data. class;
cin > > data. roll;
cin > > data. address. houseno;
cin > > data. address. street;
cin > > data. address. city;
cin > > data. address. state;
cin > > data. address. zip;
```

11.

```
cin > > person. N;
cin > > person. dob. dd;
cin > > person. dob. mm;
cin > > person. dob. yy;
```

12. (a) 100

    (b) The structures are equal

In-Text Questions 12.2

1. To define new data type of existing data type.

2. To make the program readable.

3. Here instead of int data type, integer can be used in a program.

In-text Questions 12.3

1. True

2. sun = 0, mon = 1, tue = 2

3. No

4. (a) enum months
   { jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec }

   (b) months m1, m2;
   m1 = jan;
   m2 = mar;

5. 0, 1, 2

6. enum colour {pink = 33, red = 45, green = 58, blue = 78 };