Notes

17

STRUCTURE, TYPEDEF & ENUMERATED DATA TYPE

In the previous lesson you have learnt that array is a collection of values of same data type. Sometimes it is required to store information, which can be of same or different types. For this purpose, structure statement is used in C++ programming language. In this lesson you will learn three important statements structure, typedef and enumerated data type of C++ programming language.

OBJECTIVES

After reading this lesson, you will 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.

17.1 STRUCTURE

A structure is a collection of simple variables which can be of same or different types. It is a user defined datatype which allows you to combine different data types to store a particular type of record. The data items in a structure are called the members of the structure. Consider the following example:

Example 1

```
# 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 <<"\n";
student S2 = {"ANKIT", 32};
cout << S2. name << S2. marks <<\n";
S3 = S2; cout << S3. name << S3. marks;
}
```

Output of the above program is

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).

Accessing of data members

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

structure variable. member name

Initialization of structure variable

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

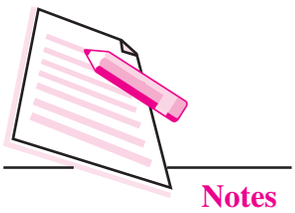
Student S2 = {"ANKITA", 32};

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.

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



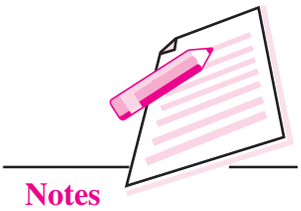
Notes

17.1.1 Structure within Structure

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

Example 2

```
# 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";
}
```



Accessing nested structure members

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

cin >> b. date_of_birth. month;

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.

17.2 TYPEDEF

It is used to define a new name 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 better understanding) for the standard data type.

The general format is:

typedef existing datatype new datatype;

For example:

typedef float real;

Now, in a program you can use datatype real instead of float. You can declare an amount variable of type real by using the statement **real amount**;

INTEXT QUESTIONS 17.1

Fill in the blanks:

1. Structure is a collection of simple variables which can be of .....
2. Typedef is used to define ..... for an existing data type.
3. ..... is the syntax of accessing data members of the structure.

4. 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";
```

17.3 ENUMERATED DATA TYPE

Enumerated data type works if you know in advance a finite list of values that a data type can take. It has the following features:

- It is a user defined datatype.
- 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 which are 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.

Example 3

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.

Enum color { red, green = 5, blue };

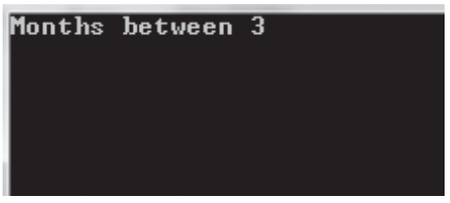
The value of blue will be 6

An enum variable takes only one value out of the possible values.

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



Notes

The members of the enumerated series is stored as integer constant in the memory. This is the reason our program displays value 3 (m2 = apr i.e., m2 = 4, m1 = jan i.e., m1 = 1. so diff = m2 – m1 = 4 – 1 = 3).

INTEXT QUESTIONS 17.2

1 What is the meaning of the following statement?

enum days { sun, mon, tue };

2 Can the enum data input by the user?

3 (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.

4 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.

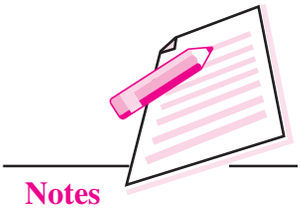
WHAT YOU HAVE LEARNT

- Structure is a collection of simple variables which can be of same or different types.
- Typedef is used to define new data type for an existing data type.
- Typedef provides alternative name for standard data type.
- Enumerated data type is a user defined data type, it works if you know in advance a finite set of values that a data type can take.
- Enum specifier defines the set of names which are stored internally as integer constant.

TERMINAL EXERCISE

1. What is a structure? Write a structure specification in C++ for the record given below:

Code: A string of 5 characters (including null)



Notes

Cost: Integer
Margin: Integer
Call this structure item.

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.

ANSWERS TO INTEXT QUESTIONS

17.1

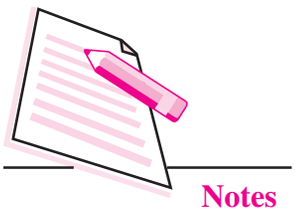
1 is a collection of simple variables which can be of same or different types.

2 new data type

3. structure variable. member

4. (a) 100

   (b) The structures are equal



Notes

17.2

1. sun = 0, mon = 1, tue = 2 2. No
3. (a) enum months
   { jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec }
   (b) months m1, m2;
   m1 = jan;
   m2 = mar;
4. enum colour {pink = 33, red = 45, green = 58, blue = 78 };