

# Structure and Pointers

## Introduction

In class XI text book, the learners have studied the method of developing C++ programs using fundamental data types and then using derived data type. This chapter starts with the discussion about structure, in which more than one data item can be grouped into a single unit. On the completion of this chapter the learner will be able to develop programs using user defined data type which can represent an entity like student, employee, book etc and hence the real world can be modelled in problem solving.

In the second part of this chapter, the use of pointer is to be discussed by introducing two memory allocation methods. After the completion of pointer section, the learner will understand the need of pointer in making effective use of computer memory and hence increasing program execution speed. Teacher can conduct a general discussion on the concept of pointer. At the end, each learner should be able to develop small programs and debug the same.

## **Values and Attitudes**

- ✓ Minimise steps in problem solving by grouping related facts and figures.
- ✓ Optimal use of available resources in real life.
- ✓ Ability to use resources based on requirements only.

## Unit Frame

Periods: 25

<p>Structure definition</p> <ul style="list-style-type: none"> <li>• Identifying</li> <li>• Explaining</li> <li>• Communicating</li> </ul>	<p>General discussion on the concept of structure by introducing general format of structure definition.</p> <ul style="list-style-type: none"> <li>• Preparation of note</li> <li>• Program coding</li> </ul> <p><i>Assessment:</i></p> <ul style="list-style-type: none"> <li>• Program code</li> </ul>	<p>1. Identifies the need of user defined data types and use structure to represent grouped data.</p>
<p>Accessing elements of structure</p> <ul style="list-style-type: none"> <li>• Identifying</li> <li>• Observing</li> </ul>	<p>General discussion by demonstrating the method of accessing structure elements using structure variables.</p> <ul style="list-style-type: none"> <li>• Program coding</li> </ul> <p>Lab work to familiarize the execution of a simple program which access structure elements.</p> <p><i>Assessment:</i></p> <ul style="list-style-type: none"> <li>• Program code</li> <li>• Lab work</li> </ul>	<p>2. Creates structure data types and accesses elements to refer to the data items.</p>
<p>Nested structure</p> <ul style="list-style-type: none"> <li>• Analyzing</li> <li>• Communicating</li> <li>• Problem solving</li> </ul>	<p>Group discussion on defining a structure which contains another structure inside in it.</p> <ul style="list-style-type: none"> <li>• Program coding</li> <li>• Error correction</li> </ul> <p>Lab work to familiarize the concept of nested data structure.</p> <p><i>Assessment:</i></p> <ul style="list-style-type: none"> <li>• Program code</li> <li>• Error correction</li> </ul>	<p>3. Use nested structure to represent data consisting of elementary data items and grouped data items.</p>
<p>Develop programs using structure</p> <ul style="list-style-type: none"> <li>• Identifying</li> <li>• Explaining</li> </ul>	<p>Discussion and demonstration of program using structure.</p> <ul style="list-style-type: none"> <li>• Program coding</li> </ul> <p>Lab work to familiarize the execution of program.</p>	<p>4. Develops C++ programs using structure data types to solve real life problems.</p>

	<i>Assessment:</i> <ul style="list-style-type: none"> <li>• Program code</li> <li>• Lab work</li> </ul>	
Pointer declaration <ul style="list-style-type: none"> <li>• Observing</li> <li>• Identifying</li> <li>• Communicating</li> </ul>	General discussion on the concept of pointer and illustrate the declaration and use of pointer variables. <ul style="list-style-type: none"> <li>• Note preparation.</li> <li>• Program coding and error correction.</li> </ul> <i>Assessment:</i> <ul style="list-style-type: none"> <li>• Program code</li> <li>• Predicted output</li> <li>• Error correction</li> </ul>	5. Explains the concept of pointers and uses pointer with the operators & and *.
Methods of memory allocation. <ul style="list-style-type: none"> <li>• Comparing</li> <li>• Communicating</li> <li>• Inferring</li> </ul>	General discussion and comparison on two methods of memory allocation <ul style="list-style-type: none"> <li>• Note preparation</li> <li>• Program coding</li> </ul> <i>Assessment:</i> <ul style="list-style-type: none"> <li>• Program code</li> </ul>	6. Compares the two types of memory allocation and use dynamic operators new and delete.
Operations on pointers <ul style="list-style-type: none"> <li>• Observing</li> <li>• Identifying errors</li> <li>• Predicting</li> </ul>	Discussion and demonstration of operation on pointer variables. <ul style="list-style-type: none"> <li>• Output prediction</li> <li>• Error correction</li> </ul> <i>Assessment:</i> <ul style="list-style-type: none"> <li>• Program code</li> <li>• Predicted output</li> <li>• Error correction</li> </ul>	7. Illustrates the operations on pointers and predict the outputs.
Pointer and array <ul style="list-style-type: none"> <li>• Identifying</li> <li>• Predicting</li> </ul>	General discussion on the relationship between array and pointer <ul style="list-style-type: none"> <li>• Note preparation</li> <li>• Program coding</li> </ul>	8. Establishes the relationship between pointers and array.

	<ul style="list-style-type: none"> <li>• Output prediction and error correction</li> </ul> <i>Assessment:</i> <ul style="list-style-type: none"> <li>• Program code</li> <li>• Predicted output</li> <li>• Error correction</li> </ul>	
Pointer and string <ul style="list-style-type: none"> <li>• Observing</li> <li>• Identifying</li> </ul>	General discussion on the relationship between string and pointer. <ul style="list-style-type: none"> <li>• Note preparation</li> <li>• Program coding</li> <li>• Output prediction and error correction</li> </ul> <i>Assessment:</i> <ul style="list-style-type: none"> <li>• Program code</li> <li>• Predicted output</li> <li>• Error correction</li> </ul>	9. Uses pointers to handle strings.
Self referential structure <ul style="list-style-type: none"> <li>• Identifying</li> <li>• Communicating</li> <li>• Inferring</li> </ul>	General discussion on the concept of self referential structure. <ul style="list-style-type: none"> <li>• Note preparation.</li> </ul> <i>Assessment:</i> <ul style="list-style-type: none"> <li>• Activity log</li> </ul>	10. Explains the concept of self referential structure.

## Towards the Unit:

(1 Period)

### User defined data type and structure

**Suggested Activity:** Discussion, Demonstration and Problem Solving.

- Teacher illustrates the importance of user defined data type by discussing the points like:
  - o Fundamental data type can represent only elementary data item.
  - o Classification of data types into three in class XI text book.
  - o Derived data type is discussed through the concept of array in class XI.
  - o Need of data type to represent compound data like student details, employee details, ... which is the user defined data types
- Teacher introduces the concept of structure along with definition syntax.
- Students are asked to demonstrate or define structure for given entity and they are also asked to write program to access structure member.
- Teacher consolidate the discussion after ensuring participation of students in programming/problem solving.

### Sample Questions

1. *LO: 1.1*                      *Type: Objective*    *Score: 1*

Structure is a \_\_\_\_\_ type of data type.

Fill up the blank by choosing correct option from the following.

- |                 |             |
|-----------------|-------------|
| a. Fundamental  | b. Derived  |
| c. User defined | d. Built in |

Processing skill category CA

Scoring indicator User defined

2. *LO: 1.3*                      *Type: Short answer*    *Score: 3*

Define a structure to represent employee details. Employee details are employee code, employee name, date of joining and place of employee.

Processing skill category: CG

Scoring indicator: Define a nested structure

3. *LO: 1.4*                      *Type: Essay*    *Score: 5*

Develop a program using a structure to read details of 50 employees (Details are employee code, employee name and salary) and display details of every employee after incrementing salary by 35%.

Processing skill category: CA

Scoring indicator: Structure definition and correct code to access the structure elements

4. *LO: 1.5*                      *Type: Short answer*    *Score: 2*

Consider the following C++ program code.

```
int a=5,*p;  
p=a;  
cout<<p;
```

Identify the error in the above program segment and correct the error.

Processing skill category: CA

Scoring indicator : p=a; Pointer variable can store only address

5. *LO: 1.5*                      *Type: Short answer*    *Score: 2*

Correct the program code given below:

```
int a=5;  
float *p;  
p=&a;  
cout<<p;
```

Processing skill category: CA

Scoring indicator : float \*p ; is to be corrected as int \*p;

6. *LO: 1.7*                      *Type: Objective*    *Score: 1*

Assume that x is a variable at address 1000.

```
float x=4.5,*ptr;  
ptr=&x;  
cout<<ptr++<<"\t";  
cout<<<<ptr;
```

Choose the correct output for the above program segment.

- a) 1000,1002                      b) 1000,1004  
c) 1002,1004                      d) 1002,1000

Processing skill category: CG

Scoring indicator : b

7. *LO: 1.2*                      *Type: Objective*    *Score: 1*

Calculate the memory requirement for the following structure variable s1.

```
struct student
{
    int rno;
    char name[25];
    float height;
}s1;
```

Processing skill category: CA

Scoring indicator: 33 bytes

8. *LO: 1.6*                      *Type: Short answer*    *Score: 3*

Raju used four new operators and three delete operators in a program. Name the situation that is going to face and explain the situation.

Processing skill category: CA

Scoring indicator: Memory leak. Explanation.

9. *LO: 1.3*                      *Type: Short answer*    *Score: 3*

Explain the type of structure defined below and calculate the number of bytes required(in gcc) to store the structure variable s1.

```
struct student
{
    int rno,mark;
    char name[25];
    struct date
    {
        int dd,mm,yy;
    }doj;
}s1;
```

Processing skill category: CG

Scoring indicator: Definition of Nested structure. 45bytes

10. *LO: 1.2*                      *Type: Short answer*    *Score: 2*  
Write the declaration statement of a student structure variable named 's' which is initialized with admission number 19, name as 'sunil' and mark as 75.  
Processing skill category: CA  
Scoring indicator: student s = {19,"sunil",75};
11. *LO: 1.2*                      *Type: Objective*    *Score: 1*  
Choose the right answer from the list  
Which is the structure member operator? (\* ? + .)  
Processing skill category: CA  
Scoring indicator: . (Dot)
12. *LO: 1.6*                      *Type: Short answer*    *Score: 2*  
Write a function definition to swap values of two integer variables using pointer.  
Processing skill category: CA  
Scoring indicator: Correction function definition using pointer variables
13. *LO: 1.6*                      *Type: Short answer*    *Score: 3*  
Write C++ statement for the following.  
i) To declare an integer variable named 'x' using new operator.  
ii) To initialize the integer pointer variable x with value 5.  
iii) To declare a dynamic array of ten integers named x.  
Processing skill category: CG  
Scoring indicator :
- a) int \*x=new int;  
b) int \*x=new int(5);  
c) int \*x=new int[10];
14. *LO: 1.10*                      *Type: Short answer*    *Score: 2*  
"Self referential structure can be used for implementing linked list".  
Justify the statement.  
Processing skill category: CA



Scoring indicator: Correct justification/Self referential structure contains a pointer to itself.

15. *LO: 1.4*                      *Type: Essay*    *Score: 5*

Write a program to read admission no, name, mark1, mark2 and mark3 of students in a class room. Then calculate total , average and grade obtained by each student [average  $\geq 90$  then grade is A+. ,  $\geq 80$  then grade is A and so on.]. Then display adm. no., name and grade of all the students.

Processing skill category: CG

Scoring indicator: Program with array of structure variable

16. *LO: 1.8*                      *Type: Short answer*    *Score: 2*

Consider the array declaration. Assume that the first element of array is stored in 2020 location

```
int a[] = {2, 5, 8, 9, 4}, *p1, *p2;
p1 = a;
p2 = &a[0];
cout << p1 << "\t" << p2;
```

Predict the output and justify.

Processing skill category: CA

Scoring indicator : Array name itself is a pointer which stores address of first element of an array.

17. *LO: 1.10*                      *Type: Short answer*    *Score: 2*

"The dot(.) operator cannot be used for accessing structure member by using structure pointer variable " Say True or False. Justify your answer.

Processing skill category: CG

Scoring indicator: Arrow (->) operator is used for accessing structure member by using structure pointer variable.

18. *LO 1.6*                      *Type: Short answer*    *Score: 3*

Differentiate between static and dynamic memory allocation with an example.

Processing skill category: CA

Scoring indicator: Dynamic allocation during program execution and static allocation during program compilation.

<code>int a[10]</code>	static array
<code>int * a= new int[10]</code>	dynamic array.

### Info Box

- 1) In Turbo C++ and in some other compilers, the following array declaration is incorrect

```
int n;  
cin>>n;  
int a[n];
```

Because array size must be a constant value.

But in Gcc we can see that such an array declaration, in which array size is a variable which is accepted during program execution, is correct. Our text book mentioned that `int a[n]` is wrong. But it is valid in Gcc As per static memory allocation concept, this declaration is wrong.

- 2) Consider the following code.

```
char ch, *p;  
p=&ch;  
cout<<p;
```

will not display the address of character variable 'ch'.

But we know that

```
int a, *p;  
p=&a;  
cout<<p;
```

will display address of integer variable a.

Hence we can conclude that in the case of character data type, instead of displaying address, the content of address will be displayed. If it is character array the display terminates only when null character is encountered.