Chapter 1

Structure and pointer

Structure

It is a userdefined datatype

It is a collection of heterogenous items

It can be defined using the keyword struct

Structure is a user defined data type of c++ to represent the collection of logically related data items which maybe different type under a common name.

Array allows to define type of variable that can hold several data itemsof the same kind.similarly structure is the another user defined datatype available in c++ that allows to combine data items of different kind.

Syntax:

Struct struct tagname

{

Datatype variable1;

Datatype variable 2;

Datatype variable n;

};

Structure end with semicolon

Example of structure

Struct student

{

Int add\_no;

Char name[20];

Float fee;

};

Define a dtructure named ‘time’ with elements hour min and sec

Struct time

{

Int hour;

Int min;

Int sec;

};

Variable declaration and memory allocation

Variable declaration

Syntax:

Struct structname struct variable:

Or

Struct name variable;

Eg of variable declaration

Struct students;

Or

Students;

Memory allocation

Eg

Struct student;

{

Int addno;

Char name[5];

Float fee;

}

Accessing elements of structure

Array is a collection of elements and these elements are accessed using the subscript

Structure is also a collection of elemets are c++ allows the accessibility of the element individually.the period (.)is provided as the operator for this purpose and it is named dot operator

Syntax:

Structure variable name.element name;

Eg:

s.add no;

variable initialisation

syntax

structname variablename{value1,value2,value3}

eg:

student s. {122 “manu” 7.22};

or

s. addno.75;

s. name=”Manu”;

write a prgm to read and print a student detail

#include <iostream.h>

Using namespace std;

Void main()

{

Char name[5];

Int age;

Char add[10];

Cout<< “enter your name”;

Cin >> name;

Cout<<”enter your age”;

Cin>>age;

Cout<<”enter your add”;

Cin>>add;

Cout<< ”your name is”<<name<<\n”;

Cout<< your age is” <<age<<\n”;

Cout <<your add is” <<add<<\n”;

Return 0;

}

Identify and correct the error in following code

Struct

{

Int regno;

Charname[20];

Float markone=100;

}

Ans: struct student

{

Int regno;

Char name[20];

Float mark\_one;

};

Nested structure

An element of a structure may itself be another structure is called nested structure.

Syntax

Struct student struct teacher

{ {

Int reg no; int mark;

Int age; char name[5];

}; struct student

Struct teacher {

{ int regno;

Int mark; int age;

Char name[5]; }5;

Student 5; float avg;

Float avg; };

};

Initialize nested structure variable

Outer structure variable.inner structure variable.element

Array vs structure

|  |  |
| --- | --- |
| array | structure |
| Derived datatype | User defined datatype |
| Collection of same type of data | Collection of different type of data |
| Elements of an array referenced using the corresponding subscripts | Elements of an array referenced using the corresponding dot operator |
| An array become another array->multidimensional array | A structure become another structure->nested structure |

Pointer

Derived datatype

It contain memory address of another variables

It can be used to access and manipulate data stored in the memory

Pointers are only incremented or decremented.no arithmetic operations are performed

Declaring point variable

Datatype \*pointer variable name;

Eg:

Int \*p;

Char\*c;

Int \*ptr1;

Struct student \*ptr;

Benefits of pointer

More efficient in handling arrays

Pointers support dynamic memory management

Pointer increases the execution speed

Pointer reduced length and complexity of a prgam

Value at operator,indirection,operation dereference operator

Int a=5 L value is the address of the variable 1000

5 a R value is the content 5

Address of operator

The address of a variable can be assigned to pointer variable using.

Used to get the address of a variable

Eg:

Int \*p;

Int a=5;

P=a;

Memory allocation

Static memory allocation

Memory allocation before the execution of the program

Dynamic memory allocation

Memory allocation during the execution of the program

Dynamic operators

New :used to allocate memory during the execution of the program

Syntax:

Pointer variable=new datatype;

Eg:

Int \*p;

P=new int;

P=new int(5);

Delete: used to deallocate or release the memory created by new operator.

Syntax:

Delete pointer\_variable;

Eg:

Delete p;

Memory leak/orphaned memory block

If the memory created by the new operator is not removed by the delete operator then this memory willnot be re allocated.this situation known as memory leak.

Reasons:

Forgetting to delete the memory that has been allocated dynamically

Failing to execute the delete statement due to poor logic of the program code

Difference btw static and dynamic memory allocation

|  |  |
| --- | --- |
| Static memory allocation | Dynamic memory allocation |
| Variables get allocated permanently | Variables get allocated when program unit is active |
| Allocation is done before program execution | Allocation is done during program execution |
| Less efficient | More efficient |
| No memory reusability | memory reusability |
| Eg:array | Eg:linked list |

Operations on pointers

Only addition and subtraction

Relational operations

Dynamic array

It is created during runtime

The dynamic memory allocation operator new

Syntax:

Pointer=new datatype[size]

Eg:

Ptr1=new int[5];

Pointer and structure

Struct employee

{

Int ecode;

Char ename[15];

Float salary;

};

Variable initialisation using pointer

Employee \*ptr;

Accessing element

eptr->ecode=400;

self referencial structure

one of the elements in a pointer to the same structure

struct stud

{

Int roll;

Char name[6];

Stud \*p;

};

Eg:linked list

Consider the given structure definition

Struct complex

{

Int real;

Int image;

};

Write a c++ statement to create a structure variable

Write a c++ statement to store the value 20 to the structure member real

a)complex s;

or

struct complex s;

b)s.real=20;

what is the difference the two declaration statement given below

int \*ptr=new int(10);

int \*ptr=new int[10];

these two statements are dynamic memory allocation.statement 1 indicate store dynamic integer value during runtime.second statement indicate dynamic array memory allocation.it store 10 integer value at run time.

Character pointer

Pointer using character

Syntax

Char \*name[7]={“Sunday”,”Monday”,”tue”,”wed”,”thur”,”fri”,”sat”,sun”};

Advantage of character pointer

No size specification(a string of any no.of characters can be stored)

Assignment operator(=)can be used

Optimal use of memory space

Write the following c++ code

Int a[5]={10,15,20,25,30}

Int \*p=a;

a)cout << \*(p+2);

b)cout <<\*p+3;

0+2=2->20

10+3->13

Cout<< p-2 10-2=8