Object Oriented & Programming Language

Lab#02

Table of Contents

Structure: 2
Why we need Structures? 2
Declaring a Structure 2
How to access structure members in C++? 3
Function inside Structs 4
Structs of Arrays: 5
Arrays of Structs: 7
Nested Structs 9
Pass By value and pass by reference 11

Structure:

Structure is a collection of variables under a single name. Variables can be of any type: int, float, char etc. The main difference between structure and array is that arrays are collections of the same data type and structure is a collection of variables under a single name.

Why we need Structures?

As we noticed arrays are very powerful device that allow us to group large amount s of data together under a single variable name.

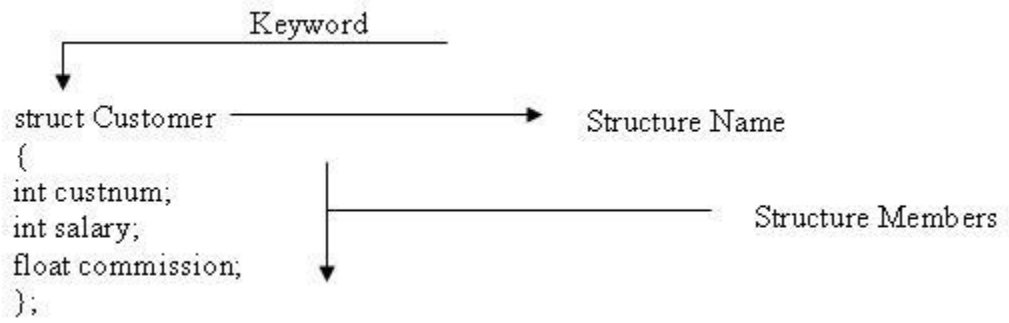
How would you write a program if, instead of being asked for simple list of either integers or characters, you were asked to combine integers, floating point numbers, and string with one variable name? You could not do it by using array.

It is actually quit often we want to group logically connected data that are of different types together. Just think about writing a program to store student record. You would need string for your name, integers to store the ID number, and a floating point number to store the grades.

Declaring a Structure

Example:

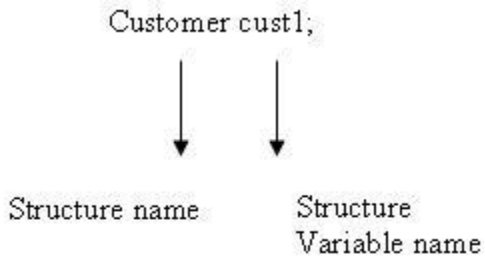
Three variables: custnum of type int, salary of type int, commission of type float are structure members and the structure name is Customer. This structure is declared as follows:



In the above example, it is seen that variables of different types such as int and float are grouped in a single structure name Customer.

Arrays behave in the same way, declaring structures does not mean that memory is allocated. Structure declaration gives a skeleton or template for the structure.

After declaring the structure, the next step is to define a structure variable.



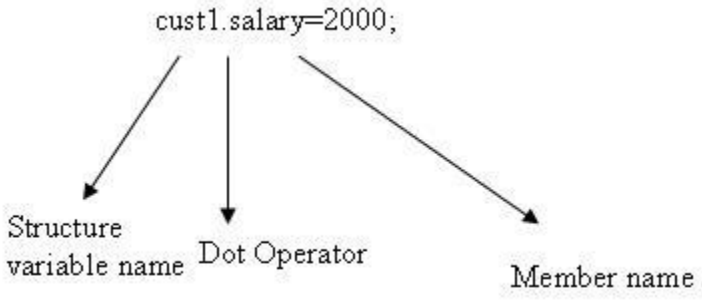
How to access structure members in C++?

To access structure members, the operator used is the dot operator denoted by (.). The dot operator for accessing structure members is used thusly:

```
Structure_Variable_name.memer_name;
```

For example:

A programmer wants to assign 2000 for the structure member salary in the above example of structure Customer with structure variable cust1 this is written as:




```
// a Complete example by declaring, using structs as well as data
members.
#include <iostream>
#include <conio.h>
#include <string>
using namespace std;
struct employe
{
string name;
double hours_worked;
double salary;
};
void main()
{
      employe a,b;
```

```
	a.name="Hifza";
	a.hours_worked=30;
	a.salary=a.hours_worked*50;

	b.name="Junaid";
	b.hours_worked=60;
	b.salary=b.hours_worked*50;

	cout<<"Employee Name:\t"<<a.name<<endl;
	cout<<"Hours worked:\t"<<a.hours_worked<<endl;
	cout<<"Salary:\t\t"<<a.salary<<endl;

	//*for second employee
	cout<<"Employee Name:\t"<<b.name<<endl;
	cout<<"Hours worked:\t"<<b.hours_worked<<endl;
	cout<<"Salary:\t\t"<<b.salary<<endl;

system("PAUSE");
}

/*Sample Output
Employee Name:  Hifza
Hours worked:   30
Salary:         1500
Employee Name:  Junaid
Hours worked:   60
Salary:         3000
Press any key to continue . . .*/
```

Function inside Structs

```
// Function inside structs
#include <iostream>
#include <conio.h>
#include <string>
using namespace std;
struct employe
{
string name;
double hours_worked;
double salary;

void input(string n, double h)
{
name=n;
hours_worked=h;
}
```

```
void print()
{
cout<<"Employee Name:\t"<<name<<endl;
cout<<"Hours worked:\t"<<hours_worked<<endl;
salary=50*hours_worked;
cout<<"Salary:\t\t"<<salary<<endl;
}
};
void main()
{
      employe a;
      a.input("ABC",10);
      a.print();
      system("PAUSE");
}
/*
Sample output:
Employee Name:  ABC
Hours worked:   10
Salary:         500

Press any key to continue . . .*/
```

Structs of Arrays:

```
#include <iostream>
#include <conio.h>
#include <string.h>
using namespace std;
struct STUDENT
{
int idNum;
int testScores[3];
int finalExam;
void init()
{
idNum=0;
testScores[0]=0;
testScores[1]=0;
testScores[2]=0;
finalExam=0;
}
void init(int a, int b[], int c)
{
idNum=a;
```

```
testScores[0]=b[0];
testScores[1]=b[1];
testScores[2]=b[2];
finalExam=c;
}
void printStudent()
{
cout<<" ID : "<<idNum
<<"\tSrcor-1 : "<<testScores[0]
<<"\tSrcor-2 : "<<testScores[1]
<<"\tSrcor-3 : "<<testScores[2]
<<"Finale : "<<finalExam<<endl<<" ";
}
};
void main()
{
STUDENT s1,s2;
// initilizing s1 elements one by one without function
s1.idNum=111;
s1.testScores[0]=95;
s1.testScores[1]=80;
s1.testScores[2]=98;
s1.finalExam=100;
s1.printStudent();
system("PAUSE");
}
/*
 ID : 111       Srcor-1 : 95    Srcor-2 : 80    Srcor-3 : 98Finale :
100
 Press any key to continue . . .
*/
```

```
#include <iostream>
#include <conio.h>
#include <string>
using namespace std;
struct STUDENT
{
int idNum;
int testScores[3];
int finalExam;
void init()
{
idNum=0;
```

```
testScores[0]=0;
testScores[1]=0;
testScores[2]=0;
finalExam=0;
}
void init(int a, int b[], int c)
{
idNum=a;
testScores[0]=b[0];
testScores[1]=b[1];
testScores[2]=b[2];
finalExam=c;
}
void printStudent(string sName)
{
cout<<"Student : "<<sName
      <<" ID : "<<idNum
<<"\tSrcors : "<<testScores[0]
<<", "<<testScores[1]
<<", "<<testScores[2]
<<"Final : "<<finalExam<<endl;
}
};
void main()
{
STUDENT s1,s2;
s1.init();
s1.printStudent("S1");
int scores[]={80,55,78};
s2.init(123,scores,98);
s2.printStudent("S2");
system("PAUSE");
}
/*
Student : S1 ID : 0     Srcors : 0, 0, 0Final : 0
Student : S2 ID : 123   Srcors : 80, 55, 78Final : 98
Press any key to continue . . .

*/
```

Arrays of Structs:

```
#include <iostream>
```

```
#include <conio.h>
#include <string>
using namespace std;
struct STUDENT
{
string sName;
int idNum;
float cgpa;
void init()
{
sName="Student";
idNum=0;
cgpa=0;
}
void init(string name, int id, float c)
{
sName=name;
idNum=id;
cgpa=c;
}
void printStudent()
{
cout<<"Name : "<<sName
<<"\tID : "<<idNum
<<"\tCGPA : "<<cgpa<<endl;
}
};
void main()
{
      STUDENT s[5];
s[0].init();
s[1].init("Najeeb",1,3.67);
s[2].init("Nadeem",2,3.57);
s[3].init("Basit",3,3.60);
s[4].init("Washaq",3,3.60);
for(int i=0;i<5;i++)// printing Studetns Strcut's Array
{
s[i].printStudent();
}
system("PAUSE");

}

/*
Name : Student  ID : 0  CGPA : 0
Name : Najeeb   ID : 1  CGPA : 3.67
```

```
Name : Nadeem   ID : 2  CGPA : 3.57
Name : Basit    ID : 3  CGPA : 3.6
Name : Washaq   ID : 3  CGPA : 3.6
Press any key to continue . . .
*/
```

Nested Structs

```
#include <iostream>
#include <conio.h>
#include <string>
using namespace std;
struct COURSE
{
int code;
string title;
int crdHours;
double gradePoints;
void init(int cd, string tl, int ch,double pts)
{
code=cd;
title=tl;
crdHours=ch;
gradePoints=pts;
}
void printCourseInfo()
{
cout<<"\t"<<code<<"\t"<<title<<"\t"<<crdHours<<"\t"<<gradePoints<<endl
;
}
};
struct STUDENT
      {
string sName;
int idNum;
double cgpa;
COURSE myCourses[5];
void init(string name, int id)
{
sName=name;
idNum=id;
}
void calculateCGPA()
{
double temp1=0,temp2=0;
```

```
for(int i=0;i<5;i++)
{
temp1+=myCourses[i].gradePoints*myCourses[i].crdHours;
temp2+=myCourses[i].crdHours;
}
cgpa= (temp1/temp2);
}
void printStudent()
{
cout<<" Name : "<<sName<<"\tID : "<<idNum
<<"\tCGPA : "<<cgpa<<endl;
cout<<"\tCode\tTitle\tCrd.Hours\tGrade Points"<<endl;
for(int i=0;i<5;i++)
{
myCourses[i].printCourseInfo();
}
}
};
void main()
{
STUDENT s1,s2;
s1.init("Safia Fatima",132);
// assigning values to courses struct one by one access
s1.myCourses[0].code=102;
s1.myCourses[0].title="Into To Com";
s1.myCourses[0].crdHours=3;
s1.myCourses[0].gradePoints=3.33;
s1.myCourses[1].init(103,"Calculus-I",3,3.00);
s1.myCourses[2].init(105,"English - I",3,2.67);
s1.myCourses[3].init(205,"Linear Algebra",3,3.67);
s1.myCourses[4].init(212,"Physics - I",3,3.33);
s1.calculateCGPA();
s1.printStudent();
system("PAUSE");
}

/*
Name : Safia Fatima    ID : 132        CGPA : 3.2
        Code    Title   Crd.Hours       Grade Points
        102     Into To Com     3       3.33
        103     Calculus-I      3       3
        105     English - I     3       2.67
        205     Linear Algebra  3       3.67
        212     Physics - I     3       3.33
Press any key to continue . . .
*/
```

```
#include <iostream>
#include <conio.h>
using namespace std;
struct Distance
{
int feet;
float inches;
};
struct Room
{
Distance length;
Distance width;
};
void main()
{
Room dining; //define a room
dining.length.feet = 13; //assign values to room
dining.length.inches = 6.5;
dining.width.feet = 10;
dining.width.inches = 0.0;
//Alternative: Room dining = { {13, 6.5}, {10, 0.0} };
//convert length & width
float l = dining.length.feet + dining.length.inches/12;
float w = dining.width.feet + dining.width.inches/12;
//find area and display it
cout << "Dining room area is " << l * w<< " square feet\n" ;
system("pause");
}
/*Sample Run:
Dining room area is 135.417 square feet
*/
```

Pass By value and pass by reference

- A `struct` variable can be passed as a parameter either by value or by reference, and
- A function can return a value of type `struct`.

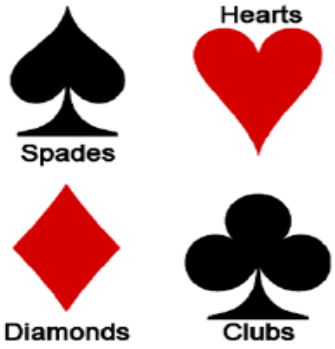
```
#include <iostream>
#include <conio.h>
using namespace std;
struct student
{
int rollno;
int examScore;
int labScore;
```

```
};
void getStudent(student& stu);
void showStudent(student stu);
student newStudent();
void main()
{
student stu1,stu2;
getStudent(stu1);
showStudent(stu1);
cout<<"\nNow with struct returning function"<<endl;
stu2 = newStudent();
showStudent(stu2);
system("pause");
}

void getStudent(student& stu)
{
cout<<"Enter Student's Information"<<endl;
cout<<"Roll No: ";
cin>>stu.rollno;
cout<<"Exam Score: ";
cin>>stu.examScore;
cout<<"Lab Score: ";
cin>>stu.labScore;
}
void showStudent(student stu)
{
cout<<"\nStudent's Informantion"<<endl;
cout<<"Roll No: "<<stu.rollno<<endl;
cout<<"Exam Score: "<<stu.examScore<<endl;
cout<<"Lab Score: "<<stu.labScore<<endl;
}
student newStudent()
{
student std1;
std1.rollno = 123;
std1.examScore = 60;
std1.labScore = 15;
return std1;
}
/*Sample Run:
Enter Student's Information
Roll No: 5
Exam Score: 40
Lab Score: 8
Student's Informantion
```

```
Roll No: 5
Exam Score: 40
Lab Score: 8
Now with struct returning function
Student's Informantion
Roll No: 123
Exam Score: 60
Lab Score: 15
*/
```

Game of cards:




```
#include <iostream>
#include <conio.h>
using namespace std;
const int clubs = 0;
const int diamonds = 1;
const int hearts = 2;
const int spades = 3;
const int jack = 11;
const int queen = 12;
const int king = 13;
const int ace = 14;
struct card
{
	int number; //2 to 10, jack, queen, king, ace
	int suit; //clubs, diamonds, hearts, spades
};

void main()
{
	card temp, chosen, prize; //define cards
	int position;

	card card1 = { 7, clubs }; //initialize card1
	cout << "Card 1 is the 7 of clubs\n";

	card card2 = { jack, hearts }; //initialize card2
	cout << "Card 2 is the jack of hearts\n";

	card card3 = { ace, spades }; //initialize card3
	cout << "Card 3 is the ace of spades\n";
```

```
	prize = card3; //copy this card, to remember it
	cout << "I'm swapping card 1 and card 3\n";
	temp = card3;
	card3 = card1;
	card1 = temp;

	cout << "I'm swapping card 2 and card 3\n";
	temp = card3;
	card3 = card2;
	card2 = temp;

	cout << "I'm swapping card 1 and card 2\n";
	temp = card2; card2 = card1; card1 = temp;

	cout << "Now, where (1, 2, or 3) is the ace of spades? ";
	cin >> position;
	switch (position)
	{
	case 1: chosen = card1; break;
	case 2: chosen = card2; break;
	case 3: chosen = card3; break;
	}
	if(chosen.number == prize.number && chosen.suit == prize.suit)
		cout << "That's right! You win!\n";
	else
		cout << "Sorry. You lose.\n";
	system("pause");
}
/*
Card 1 is the 7 of clubs
Card 2 is the jack of hearts
Card 3 is the ace of spades
I'm swapping card 1 and card 3
I'm swapping card 2 and card 3
I'm swapping card 1 and card 2
Now, where (1, 2, or 3) is the ace of spades? 2
That's right! You win!
Press any key to continue . . .
*/
```