|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | **Programming Fundamentals** | |  | **(CL214)** | |  | **LABORATORY MANUAL** | |  | **Spring 2021** | |  | **C:\Users\Aamer\Desktop\nu-new.png**  **LAB 03** | |  | **Structure** | |  | **Engr. Ibrar Khan**  **Engr. Sana Saleh**  **Muhammad Adeel Akhtar 20I-1025 B** |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_ | | | \_\_\_ | | STUDENT NAME | | ROLL NO | | | SEC | |  | | | | | | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | | | | LAB ENGINEER SIGNATURE & DATE | | | | | | | **MARKS AWARDED: /10** | | | | | | |  | | | | | | | **NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES (NUCES), ISLAMABAD** | | | | | | |  | | | | | | | Prepared by: | Engr. Sana Saleh | | Date: | 15 Feb, 2019 | | | Verified by: | Engr. Shahid Qureshi | | Date: | 16 Feb, 2019 | | |

|  |  |
| --- | --- |
| **LAB 03** | **Structure** |

**Lab Objectives:**

1. To learn what are structures in C++
2. To learn how to define a structure and access its data members.
3. To learn about array of structures.

**Software Required:**

* Dev C++

**Introduction:**

1. **Structures in C++**

Structure is the collection of variables of different types under a single name for better visualization of problem. Arrays is also collection of data but arrays can hold data of only one type whereas structure can hold data of one or more types.

## How to declare a structure in C++ Programming?

The***struct***keyword declares a structure type followed by an identifier (name of the structure). Then inside the curly braces, you can declare one or more members (declare variables inside curly braces) of that structure, termed as definition of structure. For example:

struct person {

char name[100];

int age;

float salary;

};

Here a structure **person** is declared which has three members: **name**, **age** and **salary**.

When a structure is declared, no memory is allocated. The structure definition is only the blueprint for the creation of variables. You can imagine it as a datatype. When you define an integer as below:

int foo;

The *int* specifies that, variable **foo** can hold integer element only. Similarly, structure definition only specifies that, what property a structure variable holds when it is defined.

## How to declare a structure variable?

Once you have declared a structure person as above, you can declare a structure variable (or identifier) as:

person bill;

Here, a structure variable bill is defined which is of type structure **person**. When structure variable is declared only then, the required memory is allocated by the compiler. The memory of **float** is 4 bytes, memory of **int** is 4 bytes and memory of **char** is 1 byte. Hence, 108 bytes of memory is allocated for structure variable bill.

## How to access members of a structure?

The member of structure variable is accessed using dot operator. Suppose, you want to access ageof structure variablebilland assign 50 to it. You can perform this task by using following code below:

bill.age = 50;

## Example : C++ Structure

#include <iostream>

using namespace std;

struct person { // Defining a structure

char name[50];

int age;

float salary;

};

int main() {

person p1; // Declaring a struct variable

cout << "Enter Full name: ";

cin.get(p1.name, 50);

cout << "Enter age: ";

cin >> p1.age;

cout << "Enter salary: ";

cin >> p1.salary;

cout << "\nDisplaying Information." << endl;

cout << "Name: " << p1.name << endl;

cout << "Age: " << p1.age << endl;

cout << "Salary: " << p1.salary;

return 0;

}

Here a structure person is declared which has three members. Inside **main()** function, a structure variable p1 is defined. Then, the user is asked to enter information and data entered by user is displayed.

1. **Arrays of Structures**

Since an array can contain similar elements, the combination having structures within an array is an array of structures. To declare an array of structures, you must first define a structure and then declare an array variable of that type. For example, to store addresses of 100 members of the council, you need to create an array.

Now, to declare a 100-element array of structures of type emp, we will write:

emp employees [100];

This creates 100 sets of variables that are organized as defined in the structure emp. To access a specific structure, index the structure name. For instance, to print the name of structure 8, write:

cout << employees[7].name ;

Following is an example:

#include <iostream>

#include<string>

using namespace std;

struct person { // Defining a structure

int age;

int salary;

};

int main() {

person p1[5]; // Declaring a struct variable

int i = 0;

char a[10];

for (int i = 0; i < 5; i++)

{

cout << "Enter age: ";

cin >> p1[i].age;

cout << "Enter salary: ";

cin >> p1[i].salary;

}

cout << "\nDisplaying Information.\n" << endl;

for (int i = 0; i < 5; i++)

{

cout << "Age: " << p1[i].age << endl;

cout << "Salary: " << p1[i].salary<<endl;

}

return 0;

}

**Practice Problems:**

1. Write a program for the bank, that can store customer's data. To implement this program, follow the following steps:
2. Create a structure that can store the name, account number, and balance of customers. Enter at least three customers.
3. Add a $20 amount to all customers, whose balance is less than $200. Use function for the implementation of this part.
4. Add $100 to all those customers' accounts, whose balance is more than $1000. Use function for the implementation of this part.

#include<iostream>

using namespace std;

struct person{

char name[50];

int accountnumber;

int balance;

};

int addamount(person p1[]);

int addamount1(person p1[]);

int main()

{

person p1[2];

for(int i=0;i<2;i++)

{

cout<<"enter name"<<endl;

cin>>p1[i].name;

cout<<"enter account number"<<endl;

cin>>p1[i].accountnumber;

cout<<"enter balance"<<endl;

cin>>p1[i].balance;

}

addamount(p1);

addamount1(p1);

cout<<"new values are"<<endl;

for(int i=0;i<2;i++)

{

cout<<" name"<<endl;

cout<<p1[i].name<<endl;

cout<<" account number"<<endl;

cout<<p1[i].accountnumber<<endl;

cout<<" balance"<<endl;

cout<<p1[i].balance<<endl;

}

return 0;

}

int addamount( person p1[])

{

for(int i=0;i<2;i++)

{

if(p1[i].balance<200)

p1[i].balance+=20;

}

}

int addamount1( person p1[])

{

for(int i=0;i<2;i++)

{

if(p1[i].balance>1000)

p1[i].balance+=100;

}

}

1. Write a program that can compare two dates entered by the user. Make a structure named Date to store the elements day, month and year to store the dates. If the dates are equal, display "Dates are equal" otherwise display "Dates are not equal".

#include<iostream>

using namespace std;

struct date{

int day;

int month;

int year;

};

int main()

{

date d[2];

for(int i=0;i<2;i++)

{

cout<<"enter date"<<endl;

cin>>d[i].day;

cout<<"enter month"<<endl;

cin>>d[i].month;

cout<<"enter year"<<endl;

cin>>d[i].year;

}

if (d[0].day==d[1].day&&d[0].month==d[1].month&&d[0].year==d[1].year)

{

cout<<"dates are equal"<<endl;

}

else

{

cout<<"dates are not equal"<<endl;

}

return 0;

}

1. Write a C++ program to add two fractions and display the result fraction. Your program will prompt the user to input fraction 1 and fraction 2. The numerator and denominator of each fraction are input separately by space.  See the example output below. You will need to use a C++ structure to define a fraction. The structure has two members: numerator and denominator.

Enter fraction 1(numerator denominator): 1 2

  Enter fraction 2(numerator denominator): 2 5

  Result: 9/10

#include<iostream>

using namespace std;

struct fraction{

int numerator;

int denominator;

};

int main()

{

int result,value,value1;

fraction f[2];

for (int i=0;i<2;i++)

{

if(i==0)

{

cout<<"Enter fraction 1(numerator denominator)"<<endl;;

cin>>f[i].numerator;

cout<<" "<<"/"<<endl<<" ";

cin>>f[i].denominator;

}

else if(i==1){

cout<<"Enter fraction 2(numerator denominator)"<<endl;

cin>>f[i].numerator;

cout<<" "<<"/"<<endl<<" ";

cin>>f[i].denominator;

}

}

value=(f[0].numerator\*f[1].denominator)+(f[1].numerator\*f[0].denominator);

value1=(f[0].denominator\*f[1].denominator);

cout<<"fraction is"<<endl<<value<<endl;

cout<<" "<<"/"<<endl<<" "<<value1;

return 0;

}

1. Create a FAST Library management system. It is a menu driven program that depicts the working of a library. The menu options should be:
2. Display information of each book available in the library.
3. List all books of the given author enter by the user.
4. List the title of the specified book enter by the user.
5. List the count of books in the library.
6. List the books in the order of accession numbers.
7. Exit.

To implement this program, create a structure called the library to hold the **accession number, title of the book, author name, price of the book, and a flag** indicating whether book is issued or not. Create an array of type “library” having size 5 and assign value to each field of book. Then, Check all the options of the program.

**#include<iostream>**

**using namespace std;**

**struct library{**

**char name[50];**

**int accessionnumber;**

**char title[50];**

**char authorname[50];**

**int priceofthebook;**

**int flag;**

**};**

**int display( library lib[] ,int);**

**int main()**

**{**

**library lib[5];**

**for(int i=1;i<=5;i++)**

**{**

**cout<<"enter name of book"<<endl;**

**cin>>lib[i].name;**

**cout<<"enter accession number of book"<<endl;**

**cin>>lib[i].accessionnumber;**

**cout<<"enter title of book"<<endl;**

**cin>>lib[i].title;**

**cout<<"enter the author name of this book"<<endl;**

**cin>>lib[i].authorname;**

**cout<<"enter price of book"<<endl;**

**cin>>lib[i].priceofthebook;**

**cout<<"press 1 if book is issused and press 0 if book is not issused"<<endl;**

**cin>>lib[i].flag;**

**}**

**cout<<endl<<endl;**

**cout<<"enter accession number of the book for displaying its detail"<<endl;**

**int x;**

**cin>>x;**

**display(lib,x);**

**cout<<"enter the name of author"<<endl;**

**char k;**

**cin>>k;**

**}**

**int display(library lib[],int x){**

**int z;**

**for(int i=1;i<=5;i++)**

**if(x==lib[i].accessionnumber)**

**{**

**z=i;**

**}**

**cout<<"name of book is"<<endl;**

**cout<<lib[z].name<<endl;**

**cout<<" title of book of book is"<<endl;**

**cout<<lib[z].title<<endl;**

**cout<<"the author name of this book is"<<endl;**

**cout<<lib[z].authorname<<endl;**

**cout<<"price of the book"<<endl;**

**cout<<lib[z].priceofthebook<<endl;**

**if(lib[z].flag==1)**

**{**

**cout<<"book is issused"<<endl;**

**}**

**else**

**{**

**cout<<"book is not issused"<<endl;**

**}**

**}**