NAME	**SURAJ VASANTRAO WARBHE**
ROLL NO.	**231066**
GR NO.	**21910631**
SUBJECT	**OOP**
ASSIGNMENT NO.	**03**
DATE	**11 September 2020**

****

ASSIGNMENT NO. : 3

Design a base class with name, date of birth, blood group and another base class consisting of the data members such as height and weight. Design one more base class consisting of the insurance policy number and contact address. The derived class contains the data members' telephone numbers and driving license number. (a) Write a Menu driven program to carry out the following operations. (i) Build a master table (ii) Display (iii) Insert a new entries (iv) Delete (v) Edit (vi) Search for a record.

AIM : To design a base class with name, date of birth, blood group and another base class consisting of the data members such as height and weight and one more base class consisting of the insurance policy number and contact address and then derived class contains the data members' telephone numbers and driving license number.

(a) Write a Menu driven program to carry out the following operations.

(i) Build a master table (ii) Display (iii) Insert a new entries (iv) Delete (v) Edit (vi) Search for a record.

OBJECTIVE : To understand and implement the concept of

1. Multiple Inheritance

THEORY :

1. **INHERITANCE**

In C++, inheritance is a process in which one object acquires all the properties and behaviours of its parent object automatically. In such way, we can reuse, extend or modify the attributes and behaviours which are defined in other class. In C++, the class which inherits the members of another class is called derived class and the class

whose members are inherited is called base class. The derived class is the specialized class for the base class.

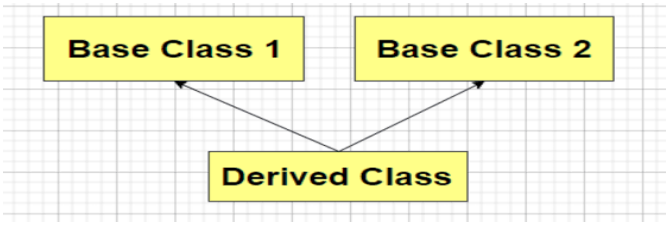
TYPES OF INHERITANCE :

There are five different types of Inheritance in C++

- Single inheritance
- Multiple inheritance
- Hierarchical inheritance
- Multilevel inheritance
- Hybrid inheritance

2. MULTIPLE INHERITANCE

Multiple Inheritance is a feature of C++ where a class can inherit from more than one classes. The constructors of inherited classes are called in the same order in which they are inherited. For example, in the following program, B's constructor is called before A's constructor.



SYNTAX

```
class
{
        //code
};
class B
{
        //code
};
class c : public A, public B(access_modifier class_name)
{
        //code
```

}

PROGRAM CODE :

```
#include <iostream>
#include <string>
using namespace std;

class A
{
  public:
        string name, dob, bg;
};

class B
{
  public:
        int ht, wt;
};

class C
{
  public:
        string ipn, add;
};

class D : public C, public B, public A
{
        long int contact, lic_no;

  public:
        void accept();
        void display();
        int insert(D d[], int n);
        void search(D d[], int n);
        void edit(D d[], int n);
        int deleted(D d[], int n);
};

void D::accept()
{
        cout << "\nEnter the following details of the person";
        cin.ignore();
        cout << "\nName : ";
        cin >> name;
```

```
	cout << "\nDate of Birth(dd/mm/yyyy) : ";
	cin >> dob;
	cout << "\nBlood Group : ";
	cin >> bg;
	cout << "\nHeight(in cm) : ";
	cin >> ht;
	cout << "\nWeight(in kg) : ";
	cin >> wt;
	cin.ignore();
	cout << "\nInsuarance Policy Number : ";
	cin >> ipn;
	cout << "\nAddress : ";
	cin >> add;
	cout << "\nContact No. : ";
	cin >> contact;
	cout << "\nLicense no. : ";
	cin >> lic_no;
}

void D::display()
{
	cout << endl << name << "\t\t" << dob << "\t\t" << bg << "\t\t" << ht << "\t\t" << wt
<< "\t\t" << ipn << "\t\t" << add << "\t\t" << contact << "\t\t" << lic_no;
}

void D::edit(D d[10], int n)
{
	char c;
	int i;
	cout << "Enter Liscence Number of the record you want to edit : ";
	cin >> c;
	for (i = 0; i < n; i++)
	{
		if (c == d[i].lic_no)
			d[i].accept();
	}
}

void D::search(D d[10], int n)
{
	char c;
	int flag = 0;
	cout << "Enter the License Number you want to search : ";
	cin >> c;
	for (int i = 0; i < n; i++)
	{
```

```
                flag = 1;
                if (c == d[i].lic_no)
                {
                        cout << "\Person Details\n";
                        cout << endl << "Name"
                                << "\t\t"
                                << "Date of Birth"
                                << "\t\t"
                                << "Blood Group"
                                << "\t\t"
                                << "Height"
                                << "\t\t"
                                << "Weight"
                                << "\t\t"
                                << "Insurance Policy No"
                                << "\t\t"
                                << "Address"
                                << "\t\t"
                                << "Contact No"
                                << "\t\t"
                                << "License no";
                        cout << endl << d[i].name << "\t\t" << d[i].dob << "\t\t" << d[i].bg <<
"\t\t" << d[i].ht << "\t\t" << d[i].wt << "\t\t" << d[i].ipn << "\t\t" << d[i].add << "\t\t" <<
d[i].contact << "\t\t" << d[i].lic_no;
                        break;
                }
        }

        if (flag == 0)
        {
                cout << "Enter Valid License Number !!!";
        }
}

int D::insert(D d[10], int n)
{
        int k;
        cout << "Enter the number of entries you want to add : ";
        cin >> k;
        for (int i = 0; i < k; i++)
        {
                d[i + n].accept();
        }
        return n + k;
}
```

```
int D::deleted(D d[10], int n)
{
	char c;
	int i, flag = 0;
	cout << "Enter the License Number of person that you want to delete : ";
	cin >> c;
	for (i = 0; i < n; i++)
	{
		flag = 1;
		if (c == d[i].lic_no)
		{
			break;
		}
	}
	if (i == n)
	{
		cout << "\nPerson Not Found !!!\n";
	}
	while (i < n)
	{
		d[i] = d[i + 1];
		i++;
	}
	cout << "\nRecord Deleted Successfully...\n";
	n--;
	return n;
}

int main()
{
	int i, n, choice;
	D d[10];
	D d1;

	do
	{
		cout << "\n\tMENU"
			<< "\n(1) Accept Data"
			<< "\n(2) Display Data"
			<< "\n(3) Insert Data"
			<< "\n(4) Edit Data"
			<< "\n(5) Delete Data"
			<< "\n(6) Search Data"
			<< "\n(7) Exit";
		cout << "\nEnter your choice : ";
		cin >> choice;
```

```
switch (choice)
{
case 1:
	cout << "\nEnter the number of entries you want to add : ";
	cin >> n;
	for (i = 0; i < n; i++)
	{
		d[i].accept();
	}
	break;

case 2:
	cout << "**********MASTER TABLE**********";
	cout << endl
		<< "Name"
		<< "\t\t"
		<< "Date of Birth"
		<< "\t\t"
		<< "Blood Group"
		<< "\t\t"
		<< "Height"
		<< "\t\t"
		<< "Weight"
		<< "\t\t"
		<< "Insurance Policy No"
		<< "\t\t"
		<< "Address"
		<< "\t\t"
		<< "Contact No"
		<< "\t\t"
		<< "License no";

	for (i = 0; i < n; i++)
	{
		d[i].display();
	}
	break;

case 3:
	n = d1.insert(d, n);
	cout << "**********MASTER TABLE**********";
	cout << endl
		<< "Name"
		<< "\t\t"
		<< "Date of Birth"
```

```
                << "\t\t"
                << "Blood Group"
                << "\t\t"
                << "Height"
                << "\t\t"
                << "Weight"
                << "\t\t"
                << "Insurance Policy no"
                << "\t\t"
                << "Address"
                << "\t\t"
                << "Contact no"
                << "\t\t"
                << "License no";
        break;

case 4:
        d1.edit(d, n);
        break;

case 5:
        n = d1.deleted(d, n);
        cout << "***********MASTER TABLE**************";
        cout << endl
                << "Name"
                << "\t\t"
                << "Date of Birth"
                << "\t\t"
                << "Blood Group"
                << "\t\t"
                << "Height"
                << "\t\t"
                << "Weight"
                << "\t\t"
                << "Insurance Policy no"
                << "\t\t"
                << "Address"
                << "\t\t"
                << "Contact no"
                << "\t\t"
                << "License no";

        for (i = 0; i < n; i++)
        {
                d[i].display();
        }
```

```
                    break;

                case 6:
                    d1.search(d, n);
                    break;
                }
            } while (choice != 7);
}
```

OUTPUT

```
      MENU
(1) Accept Data
(2) Display Data
(3) Insert Data
(4) Edit Data
(5) Delete Data
(6) Search Data
(7) Exit
Enter your choice : 1

Enter the number of entries you want to add : 2

Enter the following details of the person…

Name : Suraj
Date of Birth(dd/mm/yyyy) : 17/04/2001
Blood Group : O+
Height(in cm) : 169
Weight(in kg) : 56
Insurance Policy Number : 2012
Address : Wardha
Contact No. : 9130106674
License no. : 231066

Enter the following details of the person…

Name : Rahul
Date of Birth(dd/mm/yyyy) : 11/01/2005
Blood Group : A+
Height(in cm) : 160
Weight(in kg) : 54
Insurance Policy Number : 1204
Address : Pune
Contact No. : 9864532411
License no. : 13035

      MENU
(1) Accept Data
(2) Display Data
(3) Insert Data
(4) Edit Data
(5) Delete Data
(6) Search Data
(7) Exit
```

Enter your choice : 2

**********MASTER TABLE**********

Name	DOB	B/G	Ht.	Wt.	Insurance	Add	Contact	License
Suraj	17/04/2001	O+	169	56	2012	Wardha	9130106674	231066
Rahul	11/01/2005	A+	160	54	1204	Pune	9864532411	130035

MENU
(1) Accept Data
(2) Display Data
(3) Insert Data
(4) Edit Data
(5) Delete Data
(6) Search Data
(7) Exit
Enter your choice : 3

Enter the number of entries you want to add : 1

Enter the following details of the person

Name : Nikita
Date of Birth(dd/mm/yyyy) : 14/05/1999
Blood Group : B-
Height(in cm) : 156
Weight(in kg) : 58
Insurance Policy Number : 1223
Address : Nagpur
Contact No. : 8888823146
License no. : 46789

**********MASTER TABLE**********

Name	DOB	B/G	Ht.	Wt.	Insurance	Add	Contact	License
Suraj	17/04/2001	O+	169	56	2012	Wardha	9130106674	231066
Rahul	11/01/2005	A+	160	54	1204	Pune	9864532411	130035
Nikita	14/05/1999	B-	156	58	4689	Nagpur	8888207743	217809

MENU
(1) Accept Data
(2) Display Data
(3) Insert Data
(4) Edit Data

(5) Delete Data
(6) Search Data
(7) Exit
Enter your choice : 4

Enter License Number of the record you want to edit : 217801

Enter the following details of the person
Name : Nikita
Date of Birth(dd/mm/yyyy) : 14/05/1999
Blood Group : B-
Height(in cm) : 159
Weight(in kg) : 58
Insurance Policy Number : 2347
Address : Nagpur
Contact No. : 8888823146
License no. : 217801

MENU
(1) Accept Data
(2) Display Data
(3) Insert Data
(4) Edit Data
(5) Delete Data
(6) Search Data
(7) Exit
Enter your choice : 5

Enter License Number of the record you want to delete : 217801
Record Deleted Successfully…

MENU
(1) Accept Data
(2) Display Data
(3) Insert Data
(4) Edit Data
(5) Delete Data
(6) Search Data
(7) Exit

Enter your choice : 2

**********MASTER TABLE**********

Name	DOB	B/G	Ht.	Wt.	Insurance	Add	Contact	License
Suraj	17/04/2001	O+	169	56	2012	Wardha	9130106674	231066
Rahul	11/01/2005	A+	160	54	1204	Pune	9864532411	130035

```
      MENU
(1) Accept Data
(2) Display Data
(3) Insert Data
(4) Edit Data
(5) Delete Data
(6) Search Data
(7) Exit
Enter your choice : 6

Enter License Number of the record you want to search : 231066


Person Details :
```

Name	DOB	B/G	Ht.	Wt.	Insurance	Add	Contact	License
Suraj	17/04/2001	O+	169	56	2012	Wardha	9130106674	231066

```
      MENU
(1) Accept Data
(2) Display Data
(3) Insert Data
(4) Edit Data
(5) Delete Data
(6) Search Data
(7) Exit
Enter your choice : 7

[Program finished]
```

CONCLUSION

This assignment helps us to learn about concept of Multiple Inheritance and types of Inheritance in the programming language of 'C++'. Also, we learnt about declaration and definition of classes and object inside the program.