/*
**Title:-** Develop an object oriented program in C++ to create a database of student information system containing the following information: Name, Roll number, Class, division, Date of Birth, Blood group, Contact address, telephone number, driving license no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, Copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete.

**Roll No:-**

**Class:-SE Computer**

**Sub:-**OOPL & CGL

**Date:-08/09/2020**
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

# Program-

```cpp
#include<iostream>
#include<string.h>
using namespace std;

class person_additional_info
{
        char address[20],license[20],insurance[20];
        long int contact;

        public:
                person_additional_info() //Default constructor
                {
                        strcpy(address,"XYZ");
                strcpy(license,"XY-0000000000");
                        strcpy(insurance,"XY00000000X");
                        contact=000000000;

                }

            ~person_additional_info() //Destructor
             {
                cout<<"I am in Destructor";
             }

                friend class person;  // Declaration Friend class
};
```

```cpp
//Definition of friend class
class person
{
        char name[20], dob[10], blood[10];
        float ht,wt;
        static int count; // Static variable
        person_additional_info *pai;

        public:
                person()  //Default constructor
                {       strcpy(name,"XYZ");
                strcpy(dob,"dd/mm/yy");
                strcpy(blood,"A +");
                ht=0;
                wt=0;
                        pai=new person_additional_info;
            }
                person(person*p1) //Copy constructor
                {       strcpy(name,p1->name);
                strcpy(dob,p1->dob);
                strcpy(blood,p1->blood);
                ht=p1->ht;
                wt=p1->wt;
                        pai=new person_additional_info;
                        strcpy(pai->address,p1->pai->address);
                strcpy(pai->license,p1->pai->license);
                strcpy(pai->insurance,p1->pai->insurance);
                        pai->contact=p1->pai->contact;
            }
                static void showcount()  //Static member function
                {
                        cout<<"\nNo of records count="<<count<<"\n";
            }
                ~person()    //Destructor
            {
          cout<<"\nI am in Destructor\n";
            }
                void getdata(char name[20]);
                inline void display();   // Inline function declaration

};
void person::getdata(char name[20])
{
        strcpy(this->name,name);     //this pointer
        cout<<"\n Enter date of birth";
        cin>>dob;
```

```cpp
        cout<<"\n Enter blood group";
    cin>>blood;
        cout<<"\n Enter height";
    cin>>ht;
        cout<<"\n Enter weight";
    cin>>wt;
        cout<<"\n Enter address";
        cin>>pai->address;
        cout<<"\n Enter Licence number";
    cin>>pai->license;
        cout<<"\n Enter Insurance policy number";
    cin>>pai->insurance;
        cout<<"\n Enter Contact number";
        cin>>pai->contact;
        count++;
}
//inline function definition
void person::display()
{
        cout<<"\t"<<name;
        cout<<"\t"<<dob;
        cout<<"\t"<<blood;
        cout<<"\t"<<ht;
        cout<<"\t"<<wt;
        cout<<"\t"<<pai->address;
        cout<<"\t"<<pai->license;
        cout<<"\t"<<pai->insurance;
        cout<<"\t"<<pai->contact;
}

int person::count;  //Static variable definition

int main()
{
        person *p1,*p2;
        int ch;
        p1=new person;  //call default constructor & dynamic memory allocation
        p2=new person(p1); //call copy constructor
        cout<<"\tName";
        cout<<"\tDob";
        cout<<"\t   Blood";
        cout<<"\tHt";
        cout<<"\tWt";
        cout<<"\tAddress";
        cout<<"\tLicense";
        cout<<"\tInsurance";
```

```cpp
        cout<<"\tContact";
        cout<<endl;
        cout<<"Default Constructor Value \n";
        p1->display();
        cout<<"\n";
        cout<<"Copy Constructor Value \n";
        p2->display();
int n;
        cout<<"\nEnter how many records you want??";
        cin>>n;
        person p3[n];     //array of object
        char name[20];
        int x=0;
        do
        {
            cout<<"\nWelcome to Personal database system";
                cout<<"\n1.Enter the record";
                cout<<"\n2.Display the record";
                cout<<"\n3.Exit";
                cin>>ch;
                switch(ch)
                {
                        case 1:
                        {
                                cout<<"\nEnter the Name ";
                                cin>>name;
                                p3[x].getdata(name);
                                person::showcount();  // calls static function
                                x++;
                        break;
                        }
                        case 2:
                        {
                                cout<<"\tName";
                                cout<<"\tDob";
                                cout<<"\t   Blood";
                                cout<<"\tHt";
                                cout<<"\tWt";
                                cout<<"\tAddress";
                                cout<<"\tLicense";
                                cout<<"\tInsurance";
                                cout<<"\tContact";
                                for(int i=0;i<n;i++)
                                {
                                        cout<<"\n";
                                        p3[i].display(); //calls inline function
```

```
                        }
                        break;
                }
        }
    }while(ch!=3);
    delete p1; //dynamic memory de-allocation
    delete p2;
    return 0;

}
```

## /*Output:-

| Name | Dob | Blood | Ht | Wt | Address | License | Insurance | Contact |
|------|-----|-------|-----|-----|---------|---------|-----------|---------|
| Default Constructor Value | | | | | | | | |
| XYZ | dd/mm/yy | A + | 0 | 0 | XYZ | XY-0000000000 | XY00000000X | 0 |
| | | | | | | | | |
| Copy Constructor Value | | | | | | | | |
| XYZ | dd/mm/yy | A + | 0 | 0 | XYZ | XY-0000000000 | XY00000000X | 0 |

Enter how many records you want??2

Welcome to Personal database system

1.Enter the record
2.Display the record
3.Exit1

Enter the Name  abc

 Enter date of birth 15/5/2016

 Enter blood group o+

 Enter height 5

 Enter weight 50

 Enter address pune

 Enter Licence number jhdf87

 Enter Insurance policy number hdjsg7786

 Enter Contact number 989898989

No of records count=1

Welcome to Personal database system
1.Enter the record
2.Display the record
3.Exit 2

| Name | Dob | Blood | Ht | Wt | Address | License | Insurance | Contact |
|------|-----|-------|----|----|---------|---------|-----------|---------|
| abc | 15/5/2016 | o+ | 5 | 50 | pune | jhdf87 | hdjsg7786 | 989898989 |
| XYZ | dd/mm/yy | A + | 0 | 0 | XYZ | XY-0000000000 | XY00000000X | 0 |

Welcome to Personal database system
1.Enter the record
2.Display the record
3.Exit  3

I am in Destructor

I am in Destructor

I am in Destructor

I am in Destructor

*/