**ASSIGNMENT 1**

**STATEMENT**: 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 viz, static member functions, friend class/ friend function, this pointer, inline code and dynamic memory allocation operators-new and delete. Implement all the keywords as mentioned in the problem statement.

**AIM**:To develop a student database system using OOP concepts.

**DESCRIPTION**:Create a class ‘student’ with friend functions to read and display student information.

**OOP CONCEPT USED**:

1. **Friend function**-:A friend function of a class is defined outside that class' scope but it has the right to access all private and protected members of the class.
2. **This pointer**-:In C++, this pointer is used to represent the address of an object inside a member function.
3. **Inline function**-:C++ provides an inline functions to reduce the function call overhead. Inline function is a function that is expanded in line when it is called.
4. **Dynamic memory allocation**-:Dynamic memory allocation in C/C++ refers to performing memory allocation manually by programmer.

C++ supports these functions and also has 2 operators ‘new’ and ‘delete’ that perform the task of allocating and freeing the memory in a better and easier way.

5. **Static Member Function** -:A static member function is a special member function, which is used to access only static data members, any other normal data member cannot be accessed through static member function..

**Sourcecode:**

**#include <iostream>**

**using namespace std;**

**class studinfo;**

**class student**

**{**

**string name;**

**string cls;**

**char\* div;**

**string dob;**

**string blood\_gr;**

**static int count;**

**public:**

**int roll\_no;**

**student()**

**{**

**name=" ";**

**roll\_no = 0;**

**cls = "";**

**div = new char;**

**dob = "dd/mm/yyyy";**

**blood\_gr = " ";**

**}**

**~student()**

**{**

**delete div;**

**}**

**static int getcount()**

**{**

**return count;**

**}**

**void getData(studinfo\*);**

**void dispData(studinfo\*);**

**};**

**class studinfo**

**{**

**string address;**

**long long int \*mobno;**

**long long int \*dlno;**

**friend class student;**

**public:**

**studinfo()**

**{**

**address = " ";**

**mobno = new long long int;**

**dlno = new long long int;**

**}**

**~studinfo()**

**{**

**delete mobno;**

**delete dlno;**

**}**

**void getStudData()**

**{**

**cout<<"Enter Contact Address"<<endl;**

**cin.get();**

**getline(cin,address);**

**while(1)**

**{**

**cout<<"Enter the Telephone Number"<<endl;**

**cin>>\*mobno;**

**if(((\*mobno/100000)>10000) && (\*mobno/100000)<=100000)**

**{**

**break;**

**}**

**else**

**cout<<"Please Enter a valid telephone Number"<<endl;**

**}**

**cout<<"Enter the Driving License number:"<<endl;**

**cin>>\*dlno;**

**}**

**void dispStudData()**

**{**

**cout<<"Contact Address : "<<address<<endl;**

**cout<<"Telephone Number : "<<\*mobno<<endl;**

**cout<<"Driving License Number : "<<\*dlno<<endl;**

**}**

**};**

**inline void student:: getData(studinfo\* st)**

**{**

**cout<<"Enter Student Name: ";**

**getline(cin,name);**

**cout<<"Enter Roll Number: ";**

**cin>>roll\_no;**

**cout<<"Enter Class: ";**

**cin.get();**

**getline(cin,cls);**

**cout<<"Enter the Division: ";**

**cin>>div;**

**cout<<"Enter the Date of Birth: ";**

**cin.get();**

**getline(cin,dob);**

**cout<<"enter blood group: ";**

**t:**

**{**

**cin>>blood\_gr;**

**while(1)**

**{**

**if((blood\_gr=="o+")||(blood\_gr=="o-")||(blood\_gr=="a+")||(blood\_gr=="a-")||(blood\_gr=="ab+")||(blood\_gr=="ab-")||(blood\_gr=="b+")||(blood\_gr=="b-"))**

**{**

**break;**

**}**

**else**

**{**

**cout<<"invalid! please enter again \n";**

**goto t;**

**break;**

**}**

**}**

**}**

**st->getStudData();**

**count++;**

**cout<<"Toatal Entries = "<<student::getcount()<<endl;**

**}**

**inline void student::dispData(studinfo\* st1)**

**{**

**cout<<"Student Name: "<<name<<endl;**

**cout<<"Roll Number: "<<roll\_no<<endl;**

**cout<<"Class: "<<cls<<endl;**

**cout<<"Division: "<<div<<endl;**

**cout<<"Date of Birth: "<<dob<<endl;**

**cout<<"Blood Group: "<<blood\_gr<<endl;**

**st1->dispStudData();**

**}**

**int student::count;**

**int main()**

**{**

**student\* a[100];**

**studinfo\* b[100];**

**int n=0,c,i,roll;**

**while(1)**

**{**

**cout<<"Enter your choice: "<<endl<<endl;**

**cout<<"1.Add Data\n2.Display Data\n3.Delete Data"<<endl<<endl;**

**cin>>c;**

**cin.get();**

**switch(c)**

**{**

**case 1:**

**a[n]=new student;**

**b[n]=new studinfo;**

**a[n]->getData(b[n]);**

**n++;**

**cout<<"Data Added Successfully"<<endl<<endl;**

**break;**

**case 2:**

**if(n==0)**

**{**

**cout<<"DATABASE IS EMPTY"<<endl;**

**}**

**else**

**{**

**cout<<"----------------- STUDENT DETAILS ------------------"<<endl;**

**for(i=0;i<n;i++)**

**{**

**cout<<"----------------------------------------------------"<<endl;**

**a[i]->dispData(b[i]);**

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

**}**

**}**

**break;**

**case 3:**

**if(n<1)**

**{**

**cout<<"DATABASE IS EMPTY"<<endl;**

**}**

**else**

**{**

**cout<<"Enter the Roll number of Student to be Deleted"<<endl;**

**cin>>roll;**

**for(i=0;i<n;i++)**

**{**

**if(a[i]->roll\_no == roll)**

**{**

**while(i<n-1)**

**{**

**a[i] = a[i+1];**

**b[i] = b[i+1];**

**i++;**

**break;**

**}**

**n--;**

**}**

**else**

**{**

**cout<<"RECORD DOES NOT EXIST"<<endl;**

**}**

**}**

**cout<<"Data Deleted Successfully"<<endl<<endl;**

**}**

**break;**

**}**

**}**

**return 0;**

**}**

**Output:**

**Enter your choice:**

**1.Add Data**

**2.Display Data**

**3.Delete Data**

**1**

**Enter Student Name: suraj**

**Enter Roll Number: 221038**

**Enter Class: comp A**

**Enter the Division: A**

**Enter the Date of Birth: 16/05/2000**

**enter blood group: o+**

**Enter Contact Address**

**At/post:daund,210328**

**Enter the Telephone Number**

**7261948466**

**Enter the Driving License number:**

**17438259**

**Toatal Entries = 1**

**Data Added Successfully**

**Enter your choice:**

**1.Add Data**

**2.Display Data**

**3.Delete Data**

**1**

**Enter Student Name: Abhi**

**Enter Roll Number: 227654**

**Enter Class: etc**

**Enter the Division: B**

**Enter the Date of Birth: 23/11/2000**

**enter blood group: b+**

**Enter Contact Address**

**At/post:satara,210345**

**Enter the Telephone Number**

**9256381954**

**Enter the Driving License number:**

**23658350**

**Toatal Entries = 2**

**Data Added Successfully**

**Enter your choice:**

**1.Add Data**

**2.Display Data**

**3.Delete Data**

**2**

**----------------- STUDENT DETAILS ------------------**

**----------------------------------------------------**

**Student Name: suraj**

**Roll Number: 221038**

**Class: comp A**

**Division: A**

**Date of Birth: 16/05/2000**

**Blood Group: o+**

**Contact Address : At/post:daund,210328**

**Telephone Number : 7261948466**

**Driving License Number : 17438259**

**----------------------------------------------------**

**Student Name: Abhi**

**Roll Number: 227654**

**Class: etc**

**Division: B**

**Date of Birth: 23/11/2000**

**Blood Group: b+**

**Contact Address : At/post:satara,210345**

**Telephone Number : 9256381954**

**Driving License Number : 23658350**

**Enter your choice:**

**1.Add Data**

**2.Display Data**

**3.Delete Data**

**3**

**Enter the Roll number of Student to be Deleted**

**227654**

**RECORD DOES NOT EXIST**

**Data Deleted Successfully**

**CONCLUSION**: In this following assignment, we learned the use of friend functions and this pointer. We also learned the use of dynamic memory allocation.