**ASSIGNMENT NUMBER 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.

1. **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.

**SOURCE CODE:**

#include<iostream>

#include<string.h>

using namespace std;

class student

{

private:

char name[40],dob[15],bdg[15];

int rollno,contact;

public:

static int count;

friend class personal;

person()

{

char \* name = new char[40];

char \*dob = new char[80];

char \*bdg = new char[15];

rollno=contact=0;

}

static void recordcount()

{

cout<<"\n Total no of records :"<<count;

}

};

class personal

{

private:

char year[7],div[2],lisence[20],add[10],contact[10];

public:

void getdata(student \*obj);

void displaydata(student \*obj);

friend class student;

};

int student::count=0;

void personal::getdata(student \*obj)

{

cout<<"\n Enter Name of the Student = ";

cin.ignore();

cin.getline(obj->name,39);

cout<<"\n Enter Year = ";

cin>>this->year;

cout<<"\n Enter Division = ";

cin>>this->div;

cout<<"\n Enter Date of Birth = ";

cin>>obj->dob;

cout<<"\n Enter Blood Group = ";

cin.ignore();

cin.getline(obj->bdg,14);

cout<<"\n Enter Roll no = ";

cin>>obj->rollno;

cout<<"\n Enter Contact no. = ";

cin>>this->contact;

cout<<"\n Enter City = ";

cin>>this->add;

cout<<"\n Enter the Lisence No. = ";

cin.ignore();

cin.getline(this->lisence,19);

obj->count++;

}

void personal::displaydata(student \*obj)

{

cout<<"\nName = "<<obj->name;

cout<<"\nYear = "<<this->year;

cout<<"\nDivision = "<<this->div;

cout<<"\nRoll No. = "<<obj->rollno;

cout<<"\nDate of Birth = "<<obj->dob;

cout<<"\nBlood Group = "<<obj->bdg;

cout<<"\nContact No. = "<<this->contact;

cout<<"\nCity = "<<this->add;

cout<<"\nLisence No. = "<<this->lisence;

cout<<"\n\n\n";

}

int main()

{

personal \*p1[30];

student \*p2[30];

int n=0,ch,i;

cout<<"\n\n\n \t\t\tWELCOME TO STUDENT DATABASE! \n\n\n";

do

{

cout<<"\n\n\n Menu:-";

cout<<"\n 1.Enter Information \n 2.Display Information \n 3.Exit \n";

cout<<"\n\n\n Please enter your choice.\n";

cout<<" ";

cin>>ch;

switch(ch)

{

case 1:

cout<<"\n\t\t\t\t\t\t Enter The Information\n";

cout<<"\n";

p1[n]=new personal;

p2[n]=new student;

p1[n]->getdata(p2[n]);

n++;

student::recordcount();

break;

case 2:

cout<<"\n";

cout<<"\n---------------------------------------------------------------------------------------------------------\n";

cout<<"\n STUDENT PROFILES \n";

cout<<"\n---------------------------------------------------------------------------------------------------------\n";

cout<<"\n";

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

{

p1[i]->displaydata(p2[i]);

}

student::recordcount();

break;

case 3:

cout<<"\n\n\t\t\t\t\t\t\tThank You!\n";

break;

}

}while(ch!=3);

return 0;

}

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