

Student Registration Project using C++

```
#include<iostream>
```

```
#include<string>
```

```
#include<stdlib.h>
```

```
using namespace std;
```

```
void Add_Student();
```

```
void Display();
```

```
void Search_Student(int searchKey);
```

```
void Upadte_Student(int IDNUmbe);
```

```
void Delete(int DeleteKey);
```

```
void Sort();
```

```
    int NoStud;
```

```
struct Student
```

```
{
```

```
    string FName;
```

```
    string LName;
```

```
    int id;
```

```
    string Dep;
```

```
};
```

```
Student Stud[50];
```

```
int main()
```

```
{
```

```

int Choice=0;

cout<<"Enter the number of studnts tha u wanna regitre:";

cin>>NoStud;

while(Choice!=7){

    cout<<"\nWhat u want to do?"<<endl;

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

    cout<<"Press 1.....To Add New Student"<<endl;

    cout<<"Press 2.....to Display Student Data"<<endl;

    cout<<"Press 3.....To Search  Student Data"<<endl;

    cout<<"Press 4.....To Upadete Student Data"<<endl;

    cout<<"Press 5.....To Delete Stuent data"<<endl;

    cout<<"Press 6.....To Sotr Students based On their Id Number"<<endl;

    cout<<"Press 7.....to Exit"<<endl;

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

    cout<<"Ente Yuor CHooice"<<endl;

    cin>>Choice;

switch(Choice){

    case 1: Add_Student();

    break;

    case 2:Display();

    break;

    case 3:int serch_id;

        cout<<"Enter the id of the student tha u want to search:"<<endl;

        cin>>serch_id;

        Search_Student(serch_id);

```

```

        break;

case 4:

    int Id_to_update;

        cout<<"Enter the id of the student tha u want to Update:"<<endl;

        cin>>Id_to_update;

        Upadte_Student(Id_to_update);

        break;

case 5:

    int Id_To_Delete;

        cout<<"Enter the ID of the student that you want to delete:";

        cin>>Id_To_Delete;

        Delete(Id_To_Delete);

        break;

case 6:

        Sort();

        break;

case 7:

        exit(1);

        break;

default :

        cout<<"*****\n+++++ERROR!!Un allowable
Choice!+++++\n*****";

    }

}

}

void Add_Student()

```

```

{

    for(int i=1;i<=NoStud;i++){

        cout<<"Enter the Idedntity Card Of Student: "<<i<<endl;

        cin>>Stud[i].id;

        cout<<"Enter the Fname Of Student: "<<i<<endl;

        cin>>Stud[i].FName;

        cout<<"Enter the Lname Of Student: "<<i<<endl;

        cin>>Stud[i].LName;

        cout<<"Enter the Departiment Of Student: "<<i<<endl;

        cin>>Stud[i].Dep;

    }


    cout<<"*****\n+++++Adding
    Successfully!+++++\n*****";

}

void Display(){

    cout<<"ID"<<" "<<"FIRST NAME"<<" "<<"LAST NAME"<<"
    "<<"DEPARTMENT"<<endl;

    cout<<"_____ "<<endl;

    for(int i=1;i<=NoStud;i++){

        cout<<Stud[i].id<<"\t";

        cout<<Stud[i].FName<<"\t";

        cout<<Stud[i].LName<<"\t";

        cout<<Stud[i].Dep<<endl;

    }

}

```

```

void Search_Student(int searchKey){
    int founded_id;
    int use=0;
    for (int i=1;i<=NoStud;i++){

        if(Stud[i].id==searchKey)
        {
            use=1;
            founded_id=i;
        }
    }
    if(use==1){
        cout<<"ID"<<" "<<"FIRST NAME"<<" "<<"LAST NAME"<<"
"<<"DEPARTMENT"<<endl;

        cout<<"_____ "<<endl;

        cout<<Stud[founded_id].id<<"\t";
        cout<<Stud[founded_id].FName<<"\t";
        cout<<Stud[founded_id].LName<<"\t";
        cout<<Stud[founded_id].Dep;

    }
    else
    {
        cout<<"*****\nTher is no student With this ID, Please enter
the valid Id!\n*****";
    }
}

void Upadte_Student(int IDNUmbe)

```

```

{
    int Founded_id_to_Update;
int use=0;
for (int i=1;i<=NoStud;i++){

    if(Stud[i].id==IDNUmbe)
    {
        use=1;
        Founded_id_to_Update=i;
    }
}
if(use==1)
{
    int new_id;
    string new_FName, new_LName, New_Dep;
    cout<<"Enter the new Detail of the Student."<<endl;
    cout<<"Enter new_id Number:";
    cin>>new_id;
    cout<<"Enter new_First Name:";
    cin>>new_FName;
    cout<<"Enter new_Last Name:";
    cin>>new_LName;
    cout<<"Enter new_Departiment:";
    cin>>New_Dep;

    Stud[Founded_id_to_Update].id=new_id;

```

```

        Stud[Founded_id_to_Update].FName=new_FName;

        Stud[Founded_id_to_Update].LName=new_LName;

        Stud[Founded_id_to_Update].Dep=New_Dep;

cout<<"*****\n+++Updating
Sucesfully+++\n*****"<<endl;

    }

else

    {

cout<<"*****\n+++Error
Updating+++\n*****"<<endl;

    }

}

void Delete(int DeleteKey)

{

int Founded_id_to_Delete;

int use=0;

for (int i=1;i<=NoStud;i++){

        if(Stud[i].id==DeleteKey)

        {

                use=1;

                Founded_id_to_Delete=i;

        }

}

if(use==1)

```

```

{
    for(int k=Founded_id_to_Delete;k<=NoStud;k++)
    {
        Stud[k].id=Stud[k+1].id;

        Stud[k].FName=Stud[k+1].FName;

        Stud[k].LName=Stud[k+1].LName;

        Stud[k].Dep=Stud[k+1].Dep;
    }

    cout<<"*****\n+++Deleting
    Succesfully+++\n*****"<<endl;
}

else
{
    cout<<"*****\n+++Error Deleting, Because the is not Student With in
    this ID+++\n*****"<<endl;
}

}

void Sort()
{
    int tempId;

    string tempFName, tempLName, tempDep;

    for(int i=1; i<=NoStud;i++)
    {
        for(int j=1; j<=NoStud; j++)
        {
            if(Stud[i].id<Stud[j].id)
            {

```



```
tempId=Stud[i].id;
Stud[i].id= Stud[j].id;
Stud[j].id=tempId;
```

```
tempFName=Stud[i].FName;
Stud[i].FName=Stud[j].FName;
Stud[j].FName=tempFName;
```

```
tempLName=Stud[i].LName;
Stud[i].LName=Stud[j].LName;
Stud[j].LName=tempLName;
```

```
tempDep=Stud[i].Dep;
Stud[i].Dep=Stud[j].Dep;
Stud[j].Dep=tempDep;
```

```
}
```

```
}
```

```
}
```

```
cout<<"*****\n+++++Sorting Successfully+++++\n+++++Kow press 2 to  
see the sorted data+++++\n*****";
```

```
}
```