**Solution**

**Assignment 1**

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

class student

{

private:

int id;

float cgpa;

string name;

student \*n;

student \*start;

public:

student()

{

start = NULL;

}

void insertatend()

{

student \*t1 = new student();

if(start == NULL)

{

start = t1;

t1->n = NULL;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

else

{

student \*t = start;

while(t->n != NULL)

t=t->n;

t->n = t1;

t1->n = NULL;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

}

void insertatfront ()

{

student \*t1 =new student();

t1->n = start;

start = t1;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

void display ()

{

if (start == NULL)

cout<<"list is empty";

else

{

student \*t = start;int j=1;

while(t != NULL)

{

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

cout<<"Record No : "<<j<<endl;

cout<<"ID : "<<t->id<<endl;

cout<<"CGPA : "<<t->cgpa<<endl;

cout<<"NAME : "<<t->name<<endl;

t=t->n;

j++;

}

}

}

void deleteatend ()

{

if (start == NULL)

{

cout<<"List is empty";

}

else if (start->n == NULL)

{

start = NULL;

}

else

{

student \*t = start;

while (t->n->n != NULL)

t = t->n;

t->n = NULL;

}

}

void deleteatfront ()

{

if(start == NULL)

cout<<"List is empty :";

else

start = start->n;

}

void evennoshow ()

{

if (start == NULL)

cout<<"list is empty";

else

{

student \*t = start;int j=1;

while(t != NULL)

{

if(t->id%2==0)

{

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

cout<<"Record No : "<<j<<endl;

cout<<"ID : "<<t->id<<endl;

cout<<"CGPA : "<<t->cgpa<<endl;

cout<<"NAME : "<<t->name<<endl;

t=t->n;

j++;

}

else

{

t=t->n;

}

}

}

}

void count ()

{

int j=0;

if(start == NULL)

{

cout<<"LIST IS EMPTY"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

j++;

t=t->n;

}

cout<<"Total no of studentdent is : "<<j<<endl;

}

}

void count2 ()

{

int j=0;

if(start == NULL)

{

cout<<"LIST IS EMPTY"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->cgpa>3.0)

{

j++;

}

t=t->n;

}

cout<<"No of studentdents having cgpa>3.0 : "<<j<<endl;

}

}

void Avgcgpa ()

{

int j=0;float avg = 0;

if(start == NULL)

{

cout<<"LIST IS EMPTY"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

j++;

avg = avg+t->cgpa;

t=t->n;

}

avg = avg/j;

cout<<"Average of all studentdent cgpa is : "<<avg<<endl;

}

}

void insertafterpos ()

{

int pos;int j=0;int k=0;

cout<<"Enter the position please : ";

cin>>pos;

if(start == NULL)

{

cout<<"LIST IS EMPTY AND INVALID POSITION"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

j++;

if(j == pos)

{

student \*t1 = new student;

t1->n = t->n;

t->n = t1;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

cout<<"Node is inserted after given position."<<endl;

k=1;

break;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Node is not insert because of invalid position."<<endl;

}

}

}

void insertafterid ()

{

int id;int k=0;

cout<<"Enter the Id please : ";

cin>>id;

if(start == NULL)

{

cout<<"LIST IS EMPTY SO WE INSERT NODE AT FRONT."<<endl;

student \*t1 = new student;

start = t1;

t1->n = NULL;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->id == id)

{

student \*t1 = new student;

t1->n = t->n;

t->n = t1;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

cout<<"Node is inserted after given Id."<<endl;

k=1;

break;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Id not matched so we insert node at end."<<endl;

t = start;

while (t->n != NULL)

{

t=t->n;

}

student \*t1 = new student;

t1->n = NULL;

t->n = t1;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

}

}

void insertaftercgpa ()

{

float cgpa;int k=0;

cout<<"Enter the cgpa please : ";

cin>>cgpa;

if(start == NULL)

{

cout<<"LIST IS EMPTY SO WE INSERT NODE AT FRONT."<<endl;

student \*t1 = new student;

start = t1;

t1->n = NULL;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->cgpa == cgpa)

{

student \*t1 = new student;

t1->n = t->n;

t->n = t1;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

cout<<"Node is inserted after given cgpa."<<endl;

k=1;

break;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Cgpa not matched so we insert node at end."<<endl;

t = start;

while (t->n != NULL)

{

t=t->n;

}

student \*t1 = new student;

t1->n = NULL;

t->n = t1;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

}

}

void insertaftername ()

{

string name;int k=0;

cout<<"Enter the name please : ";

cin>>name;

if(start == NULL)

{

cout<<"LIST IS EMPTY SO WE INSERT NODE AT FRONT."<<endl;

student \*t1 = new student;

start = t1;

t1->n = NULL;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->name == name)

{

student \*t1 = new student;

t1->n = t->n;

t->n = t1;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

cout<<"Node is inserted after given nams."<<endl;

k=1;

break;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Name not matched so we insert node at end."<<endl;

t = start;

while (t->n != NULL)

{

t=t->n;

}

student \*t1 = new student;

t1->n = NULL;

t->n = t1;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

}

}

void searchbypos ()

{

int pos;int j=0;int k=0;

cout<<"Enter the position please : ";

cin>>pos;

if(start == NULL)

{

cout<<"LIST IS EMPTY AND INVALID POSITION"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

j++;

if(j == pos)

{

cout<<"Position matches and node found."<<endl;

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

cout<<"Record No : "<<j<<endl;

cout<<"ID : "<<t->id<<endl;

cout<<"CGPA : "<<t->cgpa<<endl;

cout<<"NAME : "<<t->name<<endl;

k=1;

break;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Node is not found because of invalid position."<<endl;

}

}

}

void searchbyid ()

{

int id;int j=0;int k=0;

cout<<"Enter the Id please : ";

cin>>id;

if(start == NULL)

{

cout<<"LIST IS EMPTY AND INVALID ID"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->id == id)

{

j++;

cout<<"Id matches and node found."<<endl;

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

cout<<"Record No : "<<j<<endl;

cout<<"ID : "<<t->id<<endl;

cout<<"CGPA : "<<t->cgpa<<endl;

cout<<"NAME : "<<t->name<<endl;

k=1;

break;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Node is not found because of invalid Id."<<endl;

}

}

}

void searchbycgpa ()

{

float cgpa;int j=0;int k=0;

cout<<"Enter the cgpa please : ";

cin>>cgpa;

if(start == NULL)

{

cout<<"LIST IS EMPTY AND INVALID CGPA"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->cgpa == cgpa)

{

j++;

cout<<"cgpa matches and node found."<<endl;

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

cout<<"Record No : "<<j<<endl;

cout<<"ID : "<<t->id<<endl;

cout<<"CGPA : "<<t->cgpa<<endl;

cout<<"NAME : "<<t->name<<endl;

k=1;

t=t->n;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Node is not found because of invalid cgpa."<<endl;

}

}

}

void insertatend()

{

student \*t1 = new student();

if(start == NULL)

{

start = t1;

t1->n = NULL;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

else

{

student \*t = start;

while(t->n != NULL)

t=t->n;

t->n = t1;

t1->n = NULL;

cout<<"Please Enter Id Cgpa and Name";

cin>>t1->id;

cin>>t1->cgpa;

cin>>t1->name;

}

}

void deletenodebypos ()

{

int pos;int j=0;int k=0;

cout<<"Enter the position please : ";

cin>>pos;

pos--;

if(start == NULL)

{

cout<<"LIST IS EMPTY AND INVALID POSITION"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

j++;

if(j == pos)

{

if(t->n->n!=NULL)

{

t->n=t->n->n;

cout<<"Node is deleted by given position."<<endl;

k=1;

break;

}

else

{

t->n=t->n->n;

cout<<"Node is deleted by given position."<<endl;

k=1;

break;

}

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Node is not deleted because of invalid position."<<endl;

}

}

}

void deletenodebyid ()

{

int id;int k=0;

cout<<"Enter the id please : ";

cin>>id;

if(start == NULL)

{

cout<<"LIST IS EMPTY AND INVALID ID"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->id == id)

{

student \*u = start;

while(u->n!=t)

{

u=u->n;

}

u->n = t->n;

t=t->n;

k=1;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Node is not deleted because of invalid id."<<endl;

}

}

}

void deletenodebycgpa ()

{

float cgpa;int k=0;

cout<<"Enter the cgpa please : ";

cin>>cgpa;

if(start == NULL)

{

cout<<"LIST IS EMPTY AND INVALID CGPA"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->cgpa == cgpa)

{

student \*u = start;

while(u->n!=t)

{

u=u->n;

}

u->n = t->n;

t=t->n;

k=1;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Node is not deleted because of invalid cgpa."<<endl;

}

}

}

void deletenodebyname ()

{

string name;int k=0;

cout<<"Enter the name please : ";

cin>>name;

if(start == NULL)

{

cout<<"LIST IS EMPTY AND INVALID NAME"<<endl;

}

else

{

student \* t = start;

while(t!=NULL)

{

if(t->name == name)

{

student \*u = start;

while(u->n!=t)

{

u=u->n;

}

u->n = t->n;

t=t->n;

k=1;

}

else

{

t=t->n;

}

}

if (k==0)

{

cout<<"Node is not deleted because of invalid nams."<<endl;

}

}

}

};

int \_tmain(int argc, \_TCHAR\* argv[])

{

student s;

int no;

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

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<" OPTIONS MENU ";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<"Press the following no to search\n";

cout<<"1 for insert at end \n";

cout<<"2 for insert at front \n";

cout<<"3 for delete at end \n";

cout<<"4 for delete at front \n";

cout<<"5 for dispaly \n";

cout<<"6 for even show\n";

cout<<"7 for count no of student\n";

cout<<"8 for count student having cgpa>3.0\n";

cout<<"9 for average cgpa of all student\n";

cout<<"10 for insert after given position\n";

cout<<"11 for insert after Id\n";

cout<<"12 for insert after cgpa\n";

cout<<"13 for insert after given name\n";

cout<<"14 for search by given position\n";

cout<<"15 for search by Id\n";

cout<<"16 for search by cgpa\n";

cout<<"17 for search by name\n";

cout<<"18 for delete node by given position\n";

cout<<"19 for delete no by id\n";

cout<<"20 for delete node by cgpa\n";

cout<<"21 for delete by name\n0 for exit : ";

cin>>no;

while(no!=0)

{

if(no==1)

{

s.insertatend();

}

if(no==2)

{

s.insertatfront();

}

if(no==3)

{

s.deleteatend();

}

if(no==4)

{

s.deleteatfront();

}

if(no==5)

{

s.display();

}

if(no==6)

{

s.evennoshow();

}

if(no==7)

{

s.count();

}

if(no==8)

{

s.count2();

}

if(no==9)

{

s.Avgcgpa();

}

if(no==10)

{

s.insertafterpos();

}

if(no==11)

{

s.insertafterid();

}

if(no==12)

{

s.insertaftercgpa();

}

if(no==13)

{

s.insertaftername();

}

if(no==14)

{

s.searchbypos();

}

if(no==15)

{

s.searchbyid();

}

if(no==16)

{

s.searchbycgpa();

}

if(no==17)

{

s.searchbyname();

}

if(no==18)

{

s.deletenodebypos();

}

if(no==19)

{

s.deletenodebyid();

}

if(no==20)

{

s.deletenodebycgpa();

}

if(no==21)

{

s.deletenodebyname();

}

if(no==22)

{

}

cin>>no;

}

system("pause"); return 0; }