#include<iostream>

#include<string.h>

using namespace std;

class node

{

public:

int PRN;

char name[50];

node \*next;

};

class Pinacle\_Club

{ node \*head;

public:

Pinacle\_Club();

node \*create();

void display(node \*);

void total();

node \*insert\_President();

void insert\_Member();

void insert\_Secretary();

node \*remove();

node \*concat(node \*, node \*);

~Pinacle\_Club();

};

Pinacle\_Club::Pinacle\_Club()//constructor defined

{

head = NULL;//initialize head to NULL

}

Pinacle\_Club::~Pinacle\_Club()//destructor defined

{

node \*temp, \*temp1;

temp = head->next;

delete head;

while (temp != NULL) //free the memory allocated

{

temp1 = temp->next;

delete temp;

temp = temp1;

}

}

/\*-------------------------------------------------------------------------

The Create function

-------------------------------------------------------------------------\*/

node \*Pinacle\_Club::create()

{

node \*temp=NULL, \*New;

int val, flag;

char n[50];

char ans = 'y';

flag = 1;

do

{

cout << "\nEnter the PRN of Student: ";

cin >> val;

// allocate memory to new node

cout << "\n Enter Name of Student: ";

cin >> n;

New = new node; // allocate memory for new node

if (New == NULL)

cout << "Unable to allocate memory\n";

New-> PRN = val;

strcpy(New->name, n);

New-> next = NULL;

if (flag == 1) // Executed only for the first time

{

head = New;

temp = head;

flag = 0;

}

else

{

/\*temp last keeps track of the most recently created node\*/

temp->next = New;

temp = New;

}

cout << "\n Do you want to enter more elements?(y/n)";

cin >> ans;

} while (ans == 'y' || ans == 'Y');

return head;

}

/\*-------------------------------------------------------------------------

The display function

-----------------------------------------------------------------------------\*/

void Pinacle\_Club::display(node \*head)

{

struct node \*temp;

temp = head;

if (temp == NULL)

{

cout << "\nThe list is empty\n";

return;

}

while (temp != NULL)

{

cout <<"["<< temp->PRN << ","<<temp->name<<"]";

temp = temp -> next;

}

}

/\*-------------------------------------------------------------------------

The total function

-----------------------------------------------------------------------------\*/

void Pinacle\_Club::total()

{

node \*temp;

int count = 0;

temp = head;

if (temp == NULL)

{

cout << "\nThe list is empty\n";

return;

}

while (temp != NULL)

{

count++;

temp = temp->next;

}

cout << "\n Total number of members in a club are " << count;

}

/\*-------------------------------------------------------------------------

The remove function

-----------------------------------------------------------------------------\*/

node \*Pinacle\_Club::remove()

{

node \*temp, \*prev;

int key;

prev = new node;

temp = head;

cout << "\nEnter the PRN of the node you want to delete: ";

cin >> key;

while (temp != NULL)

{

if (temp->PRN == key)//traverse till required node to delete

break; //is found

prev = temp;

temp = temp->next;

}

if (temp == NULL)

cout << "\nNode not found";

else

{

if (temp == head) //first node

head = temp->next;

else

prev->next = temp->next; //intermediate or end node

delete temp;

cout << "\nThe member is deleted\n";

}

return head;

}

/\*-------------------------------------------------------------------------

Function to insert at end

-----------------------------------------------------------------------------\*/

void Pinacle\_Club::insert\_Secretary()

{

node \*New, \*temp;

New=new node;

cout << "\nEnter The PRN of the Student: ";

cin >> New->PRN;

cout << "\nEnter The name of the Student: ";

cin >> New->name;

if (head == NULL)

head = New;

else

{

temp = head;

while (temp->next != NULL)

temp = temp->next;

temp->next = New;

New->next = NULL;

}

cout << "\nThe member is inserted\n";

}

/\*--------------------------------------------------------------------------

Function to insert after a node

-----------------------------------------------------------------------------\*/

void Pinacle\_Club::insert\_Member()

{

int key;

node \*temp, \*New;

New = new node;

cout << "\n Enter The PRN of the Student: ";

cin >> New->PRN;

cout << "\n Enter The name of the Student: ";

cin >> New->name;

if (head == NULL)

{

head = New;

}

else

{

cout << "\n Enter The PRN after which you want to insert the node: ";

cin >> key;

temp = head;

do

{

if (temp->PRN == key)

{

New->next = temp->next;

temp->next = New;

break;

}

else

temp = temp->next;

} while (temp != NULL);

}

cout << "\nThe member is inserted\n";

}

/\*-------------------------------------------------------------------------

Function to insert at the beginning

-------------------------------------------------------------------------\*/

node \*Pinacle\_Club::insert\_President()

{

node \*New, \*temp;

New = new node;

cout << "\n Enter The PRN of the Student: ";

cin >> New->PRN;

cout << "\n Enter the name of the Student: ";

cin >> New->name;

if (head == NULL)

head = New;

else

{

temp = head;

New->next = temp;

head = New;

}

cout << "\nThe member is inserted\n";

return head;

}

node \*Pinacle\_Club::concat(node \*head1, node \*head2)

{

node \*temp;

temp = head1;

while (temp->next != NULL)

temp = temp->next;

temp->next = head2;

cout << "\n The lists are concatenated";

return head1;

}

/\*------------------------------------------------------------------------

The main function

-------------------------------------------------------------------------\*/

int main()

{

Pinacle\_Club s;

int choice, ch1;

char ans = 'y';

node \*start=NULL;

node \*start1, \*start2;

start1 = NULL;

start2 = NULL;

do

{

cout << "\n1.Create";

cout << "\n2.Display Members";

cout << "\n3.Insert Member";

cout << "\n4.Delete Member";

cout << "\n5.Total Number of Members of Club";

cout << "\n6. Concatenate two Lists";

cout << "\n7.Quit";

cout << "\nEnter Your Choice ( 1-8): ";

cin >> choice;

switch (choice)

{

case 1: start=s.create();

break;

case 2: s.display(start);

break;

case 3: cout << "\nThe Members are \n";

s.display(start);

cout << "\nMenu";

cout << "\n1.Insert President";

cout<< "\n2.Insert Member";

cout << "\n3.Insert Secretary";

cout << "\nEnter your choice: ";

cin >> ch1;

switch (ch1)

{

case 1:start=s.insert\_President();

break;

case 2:s.insert\_Member();

break;

case 3:s.insert\_Secretary();

break;

default:cout << "\nInvalid choice";

}

break;

case 4:start=s.remove();

break;

case 5:s.total();

break;

case 6:cout << "\n Enter the data for first division\n";

start1=s.create();

cout << "\n Enter the data for second division\n";

start2 = s.create();

start=s.concat(start1, start2);

break;

default: cout << "\nInvalid choice";

}

cout << "\nWant to go to Main Menu?: ";

cin >> ans;

} while (ans == 'y' || ans == 'Y');

return 0;

}