PINNACLE CLUB

#include<iostream>

using namespace std;

//....................Node Structure..........................

struct Node //.....Node Structure for Singly Linked List

{

int prn;

char name[10];

Node \*Next;

}\*Head, \*Head1;

//..........Function to add\_Members() in List-01

void add\_Members()

{

int i, size;

Node \*Newnode, \*temp;

cout<<"\n How many members to add: ";

cin>>size;

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

{

//...Step 01: Allocate Memory Dynamically

Newnode = new struct Node;

//...Step02: Store data and address

cout<<"\n\t Enter PRN No. of Member: ";

cin>>Newnode->prn;

cout<<"\n\t Enter Name of Member: ";

cin>>Newnode->name;

Newnode->Next = NULL;

//....Step03: Attach New Node in SLL.

if(Head == NULL)

{

Head = Newnode;

temp = Head;

}

else

{

temp->Next = Newnode;

temp = Newnode;

}

}

}

//..........Function to Display Members() of List-01

void show\_Members()

{

struct Node \*temp;

cout<<"\n ....List-01....";

temp = Head;

cout<<"\n\n\t PRN Name";

while(temp != NULL)

{

cout<<"\n\t "<<temp->prn;

cout<<" - "<<temp->name;

temp = temp->Next;

}

}

//..........Function to Count Members() of List-01

void count\_Members()

{

Node \*temp;

int cnt = 0;

temp = Head;

while(temp != NULL)

{

cnt++;

temp = temp->Next;

}

cout<<"\n\t Total Members: "<<cnt;

}

//..........Function to Delete Member() of List-01

void delete\_member()

{

struct Node \*p, \*temp;

int key;

cout<<"\n Enter the PRN to delete Member: ";

cin>>key;

temp = Head;

while(temp != NULL)

{

if(key == temp->prn && temp == Head)

{

Head = Head->Next;

cout<<"\n\t Deleted : "<<temp->prn<<"-"<<temp->name;

delete temp;

break;

}

else if(key == temp->prn && temp != Head)

{

p->Next = temp->Next;

cout<<"\n\t Deleted : "<<temp->prn<<"-"<<temp->name;

delete temp;

break;

}

else

{

p = temp;

temp = temp->Next;

}

}

//....Display List-01 after deleting A Member.

show\_Members();

}

//..........Function to add\_Members() of List-02

void create\_List2()

{

int i, size;

Node \*Newnode, \*temp;

cout<<"\n\t ....For List-02....";

cout<<"\n How many members to add: ";

cin>>size;

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

{

//...Step 01: Allocate Memory Dynamically

Newnode = new struct Node;

//...Step02: Store data and address

cout<<"\n\t Enter PRN No. of Member: ";

cin>>Newnode->prn;

cout<<"\n\t Enter Name of Member: ";

cin>>Newnode->name;

Newnode->Next = NULL;

//....Step03: Attach New Node in SLL.

if(Head1 == NULL)

{

Head1 = Newnode;

temp = Head1;

}

else

{

temp->Next = Newnode;

temp = Newnode;

}

}

}

//..........Function to Concatenate List-01 & List-02

void join\_Lists()

{

Node \*temp;

temp = Head;

while(temp->Next != NULL)

temp = temp->Next;

temp->Next = Head1;

cout<<"\n After Concatenating Two Lists: \n ";

show\_Members();

}

//.............Main Function

int main()

{

Head = NULL;

cout<<"\n\n 1. Add the members as well as president or even secretary.";

add\_Members();

cout<<"\n\n 2. Display members as well as president or even secretary.";

show\_Members();

cout<<"\n\n 3. Compute total number of members of club.";

count\_Members();

cout<<"\n\n 4. Delete the members as well as president or even secretary.";

delete\_member();

cout<<"\n\n 5. Two linked lists exists. Concatenate two lists.";

Head1 = NULL; //....Head Node of List-02

create\_List2(); //....Create List-02

join\_Lists(); //....Join List-01 and List-02

cout<<"\n\n";

return 0;

}