Practical 7

#include <iostream>

#include <string.h>

using namespace std;

struct node

{

int prn, rollno;

char name[50];

struct node \*next;

};

class info

{

node \*sample\_node = NULL, \*head1 = NULL, \*temp1 = NULL, \*head2 = NULL, \*temp2 = NULL,\*head = NULL, \*temp = NULL;

int val\_prn, val\_roll,count\_mem,sflag=1;

char val\_name[20];

public:

node \*create();

void insertp();

void inserts();

void insertm();

void delm();

void delp();

void dels();

void display();

void count();

void reverse();

void rev(node \*p);

void concat();

};

node \*info::create()

{

node \*new\_node = new (struct node);

cout << "Enter name of student: ";

cin >> val\_name;

strcpy(new\_node->name, val\_name);

cout << "\nEnter prn no. of student:";

cin >> val\_prn;

new\_node->prn = val\_prn;

cout << "Enter student rollno: ";

cin >> val\_roll;

new\_node->rollno = val\_roll;

new\_node->next = NULL;

return new\_node;

}

void info::insertm()

{

node \*new\_node = create();

int key;

if (head == NULL)

{

head = new\_node;

}

else

{

cout << "\nEnter the roll number after which you want to insert new member \n";

cin >> key;

temp = head;

do

{

if(temp->rollno == key)

{

new\_node->next = temp->next;

temp->next = new\_node;

break;

}

else

{

temp = temp->next;

}

}

while (temp->next != NULL);

if (temp->next == NULL && sflag==1)

{

new\_node->next = temp->next;

temp->next = new\_node;

}

}

}

void info::inserts()

{

node \*new\_node = create();

if (head == NULL)

{

head = new\_node;

}

else

{

temp = head;

while (temp->next != NULL)

{

temp = temp->next;

}

temp->next = new\_node;

}

sflag=0;

}

void info::insertp()

{

node \*new\_node = create();

if (head == NULL)

{

head = new\_node;

}

else

{

temp = head;

head = new\_node;

head->next = temp->next;

}

}

void info::display()

{

if (head == NULL)

{

cout << "linklist is empty";

}

else

{

temp = head;

cout << "prn rollno NAME \n";

do

{

cout << " \n"<< temp->prn << " " << temp->rollno << " " << temp->name;

temp = temp->next;

cout<<temp;

}while (temp->next != NULL);

cout << " \n" << temp->prn << " " << temp->rollno << " " << temp->name;

}

}

void info::delm()

{

int del\_mem, flag = 0;

cout << "\n Enter the prn no. of student whose data you want to delete: ";

cin >> del\_mem;

temp = head;

if(head->prn==del\_mem){

head=head->next;

delete(temp);

return;

}

while (temp->next != NULL)

{

if (temp->prn == del\_mem)

{

sample\_node->next = temp->next;

delete (temp);

flag = 1;

}

else{

sample\_node = temp;

temp = temp->next;

}

}

if (flag == 0)

{

cout << "\nSorry memeber not deleted.";

}

}

void info::delp()

{

temp = head;

head = head->next;

delete (temp);

}

void info::dels()

{

temp = head;

if(head->next==NULL){

head=head->next;

delete(temp);

return;

}

while (temp->next != NULL)

{

sample\_node = temp;

temp = temp->next;

}

sample\_node->next = NULL;

delete (temp);

}

void info::count()

{

temp = head;

count\_mem = 0;

while (temp->next != NULL)

{

temp = temp->next;

count\_mem++;

}

count\_mem++;

cout << "Count of members is: " << count\_mem;

}

void info::concat()

{

int i,k, j;

cout << "Enter no. of members in list1: ";

cin >> k;

head = NULL;

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

{

inserts();

head1 = head;

}

head = NULL;

cout << "Enter no. of members in list2: ";

cin >> j;

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

{

inserts();

head2 = head;

}

head = NULL;

temp1 = head1;

while (temp1->next != NULL)

{

temp1 = temp1->next;

}

temp1->next = head2;

temp2 = head1;

cout << " prn rolln0 NAME \n";

while (temp2->next != NULL)

{

cout << "\n " << temp2->prn << " " << temp2->rollno << " " << temp2->name << "\n";

;

temp2 = temp2->next;

}

cout << "\n " << temp2->prn << " " << temp2->rollno << " " << temp2->name << "\n";

}

int main()

{

info sample\_node;

int num\_ch;

char ch;

do

{

cout << "\n Choice the options ";

cout << "\n 1. To insert president ";

cout << "\n 2. To insert member ";

cout << "\n 3. To insert secretary ";

cout << "\n 4. To delete president ";

cout << "\n 5. To delete member ";

cout << "\n 6. To delete secretary ";

cout << "\n 7. Count of members ";

cout << "\n 8. To display data ";

cout << "\n 9.To concatenate two strings ";

cout<<"\nEnter your choice :";

cin >> num\_ch;

switch (num\_ch)

{

case 1:

sample\_node.insertp();

break;

case 2:

sample\_node.insertm();

break;

case 3:

sample\_node.inserts();

break;

case 4:

sample\_node.delp();

break;

case 5:

sample\_node.delm();

break;

case 6:

sample\_node.dels();

break;

case 7:

sample\_node.count();

break;

case 8:

sample\_node.display();

break;

case 9:

sample\_node.concat();

break;

default:

cout << "\n unknown choice";

}

cout << "\nDo you want to continue ?(y/Y) \n";

cin >> ch;

}

while (ch == 'y' || ch == 'Y');

cout<<"\*\*\*\*\*\*\*\*\*\* End OF Programe";

return 0;

}

Output

