**/\* Program No. :**

**Aim : WAP to create a linked list which stores double and char type data separately using class templates and perform the following operations on the list - 1. Insert in end, 2. Delete from beginning, 3. Display.**

**\*/**

#include<iostream.h>

#include<conio.h>

template<class typea>

struct node

{

typea data;

node \*next;

};

template<class typea>

class list

{

node<typea> \*start,\*last;

public:

list()

{

start=NULL;

last=NULL;

}

void add(typea);

typea del();

void display() const;

};

template<class typea>

void list<typea>::add(typea no)

{

node<typea> \*newlink=new node<typea>;

newlink->data=no;

newlink->next=NULL;

if(start==NULL)

{

start=newlink;

last=newlink;

}

else

{

last->next=newlink;

last=newlink;

}

}

template<class typea>

typea list<typea>::del()

{

typea var;

node<typea> \*temp;

temp=start;

var=temp->data;

if(start==NULL)

return((typea)-9999);

else if(start==last)

{

start=NULL;

last=NULL;

}

else

start=start->next;

delete(temp);

return(var);

}

template<class typea>

void list<typea>::display() const

{

if(start==NULL)

cout<<"\n\n\t\tList is empty, No elements to display";

else

{

node<typea> \*temp;

temp=start;

while(temp!=NULL)

{

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

temp=temp->next;

}

}

}

void main()

{

int choice1;

double d;

char choice2,c;

list<double> l1;

list<char> l2;

do

{

clrscr();

cout<<"\n\n\t\t\t\tMenu\n\n\t\t1. Insert to Double List"

<<"\n\t\t2. Delete from Double List"

<<"\n\t\t3. Display Double List"

<<"\n\t\t4. Insert to Char List"

<<"\n\t\t5. Delete from Char List"

<<"\n\t\t6. Display Char List"

<<"\n\t\t7. Exit";

cout<<"\n\n\tEnter your choice (1-7) : ";

cin>>choice1;

switch(choice1)

{

case 1:do

{

cout<<"\n\t\tEnter element : ";

cin>>d;

l1.add(d);

cout<<"\n\tWant to enter more (y/n) : ";

cin>>choice2;

}while(choice2=='y'||choice2=='Y');

break;

case 2:do

{

d=l1.del();

if(d==(double)-9999)

{

cout<<"\n\n\t\tElement can't be deleted, because list is empty"

<<"\n\t\tPress any key";

choice2='n';

getch();

}

else

{

cout<<"\n\t\tElement deleted : "<<d;

cout<<"\n\tWant to delete more (y/n) : ";

cin>>choice2;

}

}while(choice2=='y'||choice2=='Y');

break;

case 3:l1.display();

cout<<"\n\n\t\t\t\tPress any key";

getch();

break;

case 4:do

{

cout<<"\n\t\tEnter element : ";

cin>>c;

l2.add(c);

cout<<"\n\tWant to enter more (y/n) : ";

cin>>choice2;

}while(choice2=='y'||choice2=='Y');

break;

case 5:do

{

c=l2.del();

if(c==(char)-9999)

{

cout<<"\n\n\t\tElement can't be deleted, because list is empty"

<<"\n\t\tPress any key";

choice2='n';

getch();

}

else

{

cout<<"\n\t\tElement deleted : "<<c;

cout<<"\n\tWant to delete more (y/n) : ";

cin>>choice2;

}

}while(choice2=='y'||choice2=='Y');

break;

case 6:l2.display();

cout<<"\n\n\t\t\t\tPress any key";

getch();

break;

case 7:break;

default:cout<<"\n\n\t\t\tIncorrect Choice"

<<"\n\n\t\tPress any key";

getch();

break;

}

}while(choice1!=7);

getch();

}

**/\***

**Name : Rohit Aggarwal**

**Roll No. : 7CS-097**

**\*/**