**Code**

#include<iostream>

using namespace std;

template <class t>

class node

{

public:

t data;

node \*next;

node(t x,node\* n=NULL)

{

data=x;

next=n;

}

};

template <class t>

class Stacktype

{

node<t> \*top;

public:

Stacktype()

{

top=NULL;

}

void push(node<t> \*);

t pop();

int isEmpty();

t topmost();

void clear();

void display();

node<t>\* createNode(t p);

};

template <class t>

int Stacktype<t>::isEmpty()

{

if(top==NULL)

return(1);

else

return(0);

}

template <class t>

void Stacktype<t>::push(node<t> \*p)

{

if(top==NULL)

{

top=p;

cout<<" Node pushed "<<endl;

}

else

{

node<t> \*temp=top;

top=p;

p->next=temp;

cout<<" Node pushed "<<endl;

}

}

template <class t>

t Stacktype<t>::pop()

{

node<t> \*temp=NULL;

if(isEmpty())

{

cout<<" Stack underflow condition "<<endl;

return 1;

}

else

{

temp=top;

top= top->next;

return temp->data;

delete temp;

}

}

template <class t>

t Stacktype<t>::topmost()

{

return top->data;

}

template <class t>

void Stacktype<t>::display()

{

cout<<" All values in the stack are "<<endl;

node<t> \*temp=top;

cout<<" ";

while(temp!=NULL)

{

cout<<temp->data<<" ";

temp=temp->next;

}

}

template <class t>

node<t>\* Stacktype<t>::createNode(t x)

{

node<t> \*temp=new node<t>(x);

return temp;

}

template <class t>

void Stacktype<t>::clear()

{

while(top!=NULL)

{

node<t> \*temp=top->next;

delete top;

top=temp;

}

}

int main()

{

Stacktype<int> s1;

node<int> \*temp;

int choice,d,a;

char ch='y';

do

{

cout<<" --------------Main Menu-------------- "<<endl;

cout<<" 1. Push "<<endl;

cout<<" 2. Pop "<<endl;

cout<<" 3. isEmpty "<<endl;

cout<<" 4. Topmost Node in the stack"<<endl;

cout<<" 5. Clear the stack "<<endl;

cout<<" 6. Display "<<endl;

cout<<" Enter choice "<<" ";

cin>>choice;

switch(choice)

{

case 1: cout<<" Enter data to be pushed ";

cin>>d;

temp=s1.createNode(d);

s1.push(temp);

s1.display();

break;

case 2: cout<<" Pop function called "<<endl;

a=s1.pop();

s1.display();

break;

case 3: if(s1.isEmpty())

cout<<" Stack is Empty "<<endl;

else

cout<<" Stack is not empty "<<endl;

break;

case 4: if(s1.isEmpty())

cout<<" Stack is Empty "<<endl;

else

{

cout<<s1.topmost();

cout<<endl;

}

break;

case 5: if(s1.isEmpty())

cout<<" Stack is Empty "<<endl;

else

{

s1.clear();

}

break;

case 6: if(s1.isEmpty())

cout<<" Stack is Empty "<<endl;

else

{

s1.display();

cout<<endl;

}

break;

default: cout<<" Choose correct option number "<<endl;

}

cout<<"\n Do you want to continue "<<" ";

cin>>ch;

}while(ch=='y');

return 0;

}



