**Assignment No : 1.3.1**

**Title : Implementation of program based on Stack**

**Name : Jayant Sanjay Patil**

**Roll No : 45**

#include<iostream.h>

#include<conio.h>

class STACK\_45

{

private:

int \*A,s,top,ele;

public:

STACK\_45(int);

void PUSH(int ele);

int POP();

void LIST\_ALL();

};

STACK\_45 :: STACK\_45(int par)

{

s=par; top=0;

A=new int[s+1];

}

void STACK\_45 :: PUSH(int ele)

{

if(top==s)

{

cout<<"\n\t Stack is Full";

return;

}

top=top+1;

A[top]=ele;

}

int STACK\_45 :: POP()

{

if(top==0)

{

cout<<"\n Stack is empty";

return NULL;

}

int ele=A[top];

top=top-1;

return ele;

}

void STACK\_45 :: LIST\_ALL()

{

cout<<"\n STACK elements are :";

if(top==0)

cout<<"STACK is empty"<<endl;

else

for(int i=top; i>=1; i--)

cout<<A[i]<<" ";

}

void MENU()

{

int opt,ele,size;

cout<<"\n Enter size of stack : ";

cin>>size;

STACK\_45 obj(size);

do

{

cout<<"\n 1.PUSH";

cout<<"\n 2.POP";

cout<<"\n 3.LIST\_ALL";

cout<<"\n 4.Exit";

cout<<"\n Enter your option : ";

cin>>opt;

switch(opt)

{

case 1:

cout<<"\n Enter Element to add : ";

cin>>ele;

obj.PUSH(ele);

break;

case 2:

int ele=obj.POP();

cout<<endl<<ele<<"is deleted";

break;

case 3:

obj.LIST\_ALL();

break;

case 4:

return;

default:

cout<<"\n Invalid option";

}

}

while(1);

}

void main()

{

int ele;

clrscr();

MENU();

getch();

}