**M. Ali. Arslan**

**19F-0348**

**Lab-03**

**Task # 01**

#include <iostream>

using namespace std;

class Node{

int data;

Node\* link;

public:

Node\* top = NULL;

bool isempty()

{

if (top == NULL)

{

return true;

}

else

{

return false;

}

}

void push(int value)

{

Node\* ptr = new Node();

ptr->data = value;

ptr->link = top;

top = ptr;

}

void pop()

{

if (isempty())

{

cout << "Stack is Empty";

}

else

{

Node\* ptr = top;

top = top->link;

delete(ptr);

}

}

void showTop()

{

if (isempty())

{

cout << "Stack is Empty";

}

else

{

cout << "Element at Top is: " << top->data;

}

}

};

class Stack

{

private:

int top;

int arr[5];

public:

Stack()

: top(-1) {

for (int i = 0; i < 5; i++)

{

arr[i] = NULL;

}

};

bool isEmpty1()

{

if (top == -1)

{

return true;

}

else

{

return false;

}

}

bool isFull()

{

if (top == 4)

{

return true;

}

else

{

return false;

}

}

void push1(int num)

{

if (!isFull())

{

top++;

arr[top] = num;

}

else

{

cout << "Cannot Push, Stack Full!" << endl;

}

}

int pop1()

{

if (isEmpty1())

{

cout << "Stack is Empty, cannot pop!" << endl;

}

else

{

int popVal = 0;

cout << "Element: " << arr[top] << " has been popped!" << endl;

popVal = arr[top];

arr[top] = NULL;

top--;

return popVal;

}

}

int count()

{

return (top + 1);

}

int peek(int pos)

{

if (pos <= 4)

{

if (!isEmpty1())

{

return arr[pos];

}

}

}

void change(int pos, int val)

{

if (pos <= 4)

{

if (!isEmpty1())

{

cout << "Element at position " << pos << " changed with value : " << val << endl;

arr[pos] = val;

}

}

}

void display1()

{

if (isEmpty1())

{

cout << "Nothing to display, stack is empty!" << endl;

}

else

{

cout << endl;

for (int i = 4; i >= 0; i--)

{

cout << "Element: " << arr[i] << endl;

}

cout << endl;

}

}

};

int main()

{

int choice;

cout << "1. LinkedList Based Stack\n" << "2. Array Based Stack\n" << "Choice: ";

cin >> choice;

switch (choice)

{

case 1:

{

Node a;

a.push(1);

a.push(2);

a.pop();

}

case 2:

{

Stack one;

one.display1();

one.push1(5);

one.push1(3);

one.push1(2);

one.display1();

one.pop1();

one.display1();

}

}

}

