

**LAB-07**

**Title: Algorithm to implement push and pop operations on stack.**

**Name : Azizul Abedin Azmi**

**ID : 2022-1-60-130**

**Section: 03**

**Course Code: CSE207**

**Course Title: (Data Structures)**

**Date: 29/04/2024**

**Course Instructor:**

**Dr. Anup Kumar Paul**

**Associate Professor**

**Department of Computer Science and Engineering**

**Source Code:**

**Node.java:**

package Lab07;

public class Node {

    int data;

    Node next;

    public Node(int data) {

        this.data = data;

        this.next = null;

    }

}

**Main.java:**

package Lab07;

public class Main {

    public static void main(String[] args) {

        StackOperations stack = new StackOperations();

        // Perform manual push and pop operations

        stack.push();

        stack.push();

        stack.push();

        stack.pop();

        stack.pop();

        stack.pop();

        stack.pop(); // This will demonstrate stack underflow

    }

}

**StackOperations.java:**

package Lab07;

import java.util.Scanner;

public class StackOperations {

    private Node top;

    private Scanner scanner;

    public StackOperations() {

        this.top = null;

        this.scanner = new Scanner(System.in);

    }

    public void push() {

        System.out.print("Enter data to push: ");

        int data = scanner.nextInt();

        Node newNode = new Node(data);

        if (isEmpty()) {

            top = newNode;

        } else {

            newNode.next = top;

            top = newNode;

        }

        System.out.println("Data pushed: " + data);

    }

    public void pop() {

        if (isEmpty()) {

            System.out.println("Stack Underflow");

        } else {

            int poppedData = top.data;

            top = top.next;

            System.out.println("Popped data: " + poppedData);

        }

    }

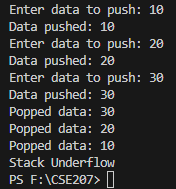
    public boolean isEmpty() {

        return top == null;

    }

}

**Output:**

****