Object Oriented Programming Lab

Lab Assignment

Submitted by:

Vimal Thomson

Roll No: 40

RMCA-B Sem-II

Submitted to:

Sr. Elsin Chakkalackal

1. Program to create a generic stack and do the Push and Pop operations.

Java code

public class StackAsLinkedList {

StackNode root;

static class StackNode {

int data;

StackNode next;

StackNode(int data) { this.data = data; }

}

public boolean isEmpty()

{

if (root == null) {

return true;

}

else

return false;

}

public void push(int data)

{

StackNode newNode = new StackNode(data);

if (root == null) {

root = newNode;

}

else {

StackNode temp = root;

root = newNode;

newNode.next = temp;

}

System.out.println(data + " pushed to stack");

}

public int pop()

{

int popped = Integer.MIN\_VALUE;

if (root == null) {

System.out.println("Stack is Empty");

}

else {

popped = root.data;

root = root.next;

}

return popped;

}

public int peek()

{

if (root == null) {

System.out.println("Stack is empty");

return Integer.MIN\_VALUE;

}

else {

return root.data;

}

}

// Driver code

public static void main(String[] args)

{

StackAsLinkedList sll = new StackAsLinkedList();

sll.push(10);

sll.push(20);

sll.push(30);

System.out.println(sll.pop()

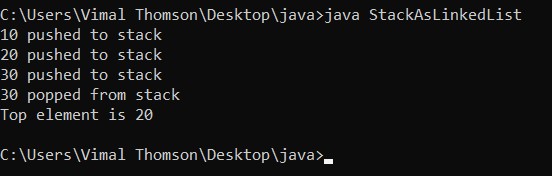
+ " popped from stack");

System.out.println("Top element is " + sll.peek());

}

}

Output



1. Using generic method perform Bubble sort.

Java code

public class BubbleSort {

static void bubbleSort(int[] arr) {

int n = arr.length;

int temp = 0;

for(int i=0; i < n; i++){

for(int j=1; j < (n-i); j++){

if(arr[j-1] > arr[j]){

temp = arr[j-1];

arr[j-1] = arr[j];

arr[j] = temp;

}

}

}

}

public static void main(String[] args) {

int arr[] ={3,60,35,2,45,320,5};

System.out.println("Array Before Bubble Sort");

for(int i=0; i < arr.length; i++){

System.out.print(arr[i] + " ");

}

System.out.println();

bubbleSort(arr);

System.out.println("Array After Bubble Sort");

for(int i=0; i < arr.length; i++){

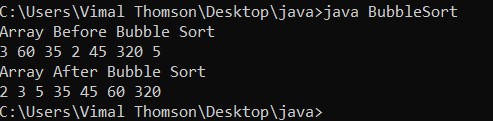
System.out.print(arr[i] + " ");

}

}

}

Output:



1. Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

Java code

import java.util.\*;

class ArrayListex{

public static void main(String args[]){

ArrayList<String> alist=new ArrayList<String>();

alist.add("Steve");

alist.add("Tim");

alist.add("Lucy");

alist.add("Pat");

alist.add("Angela");

alist.add("Tom");

//displaying elements

System.out.println(alist);

//Removing "Steve" and "Angela"

alist.remove("Steve");

alist.remove("Angela");

//displaying elements

System.out.println(alist);

//Removing 3rd element

alist.remove(2);

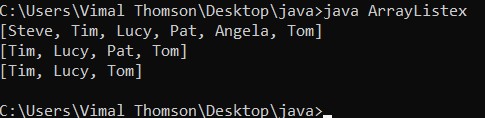
//displaying elements

System.out.println(alist);

}

}

Output



1. Program to remove all the elements from a linked list

Java code

class LinkedList

{

Node head;

class Node

{

int data;

Node next;

Node(int d) { data = d; next = null; }

}

void deleteList()

{

head = null;

}

public void push(int new\_data)

{

Node new\_node = new Node(new\_data);

new\_node.next = head;

head = new\_node;

}

public static void main(String [] args)

{

LinkedList llist = new LinkedList();

llist.push(1);

llist.push(4);

llist.push(1);

llist.push(12);

llist.push(1);

System.out.println("Deleting the list");

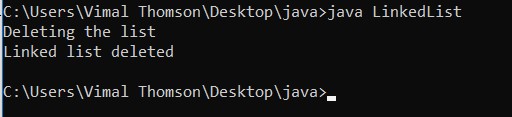
llist.deleteList();

System.out.println("Linked list deleted");

}

}

Output



1. Program to demonstrate the addition and deletion of elements in deque

Java code

import java.util.\*;

public class Dequeex {

public static void main(String[] args)

{

Deque<String> deque

= new LinkedList<String>();

deque.add("Element 1 (Tail)");

deque.addFirst("Element 2 (Head)");

deque.addLast("Element 3 (Tail)");

deque.push("Element 4 (Head)");

deque.offer("Element 5 (Tail)");

deque.offerFirst("Element 6 (Head)");

System.out.println(deque + "\n");

deque.removeFirst();

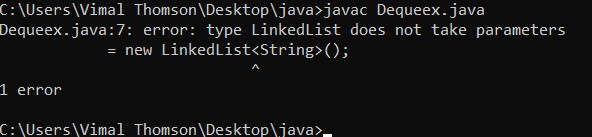
deque.removeLast();

System.out.println("Deque after removing " + "first and last: " + deque);

}

}

Output



1. Program to demonstrate the working of Map interface by adding, changing and removing elements.

Java code

import java.util.\*;

class HashMap {

public static void main(String args[])

{

Map<String, Integer> hm

= new HashMap<String, Integer>();

hm.put("a", new Integer(15));

hm.put("b", new Integer(26));

hm.put("c", new Integer(37));

hm.put("d", new Integer(48));

for (Map.Entry<String, Integer> me : hm.entrySet()) {

System.out.print(me.getKey() + ":");

System.out.println(me.getValue());

}

}

}