**PANGASINAN STATE UNIVERSITY**

 Urdaneta Campus

College of Computing

Information Technology Department

**DATA STRUCTURES AND ALGORITHM**

Stack and Queue

|  |
| --- |
|  |
|  |

Collado, Crystal Jhayne A.                                URD\_CC104\_IT Data Structures and Algorithm

BSIT – 2C                                                             Mr. Mark Dave Borromeo

**LAB ACTIVITY (STACK)**

import java.util.Stack;  public class Main {

public static void main(String[]args) {

Stack<String> stack = new Stack<>();  stack.push("Diwata");

stack.push("Robin");

stack.push("Rosmar");

stack.push("Alice");

stack.push("Beyonce");

System.out.println("Stack before pop: " + stack);

System.out.println("Top element before pop: " + stack.peek());

stack.pop();

System.out.println("Top element after pop: " + stack.peek());

System.out.println("Stack after pop: " + stack);

}

}

**LABORATORY (STACK)**

public class Main {

public static void main(String[] args) { Stack<String> stack = new Stack<>();

stack.push(3);

stack.push(2);

stack.push(0);

stack.pop();

System.out.println("Top: " + stack.top()); stack.push(1);

stack.push(4);

stack.pop();

stack.pop();

stack.pop();

System.out.println("Top: " + stack.top()); stack.push(5);

stack.push(6);

System.out.println("Top: " + stack.top());

stack.pop();

 System.out.println("Size: " + stack.size());

System.out.println("IsFull: " + stack.isFull());

System.out.println("IsEmpty: " + stack.isEmpty());

stack.display();

  }

}

**Final Output:**

* Top: 5
* Size: 2
* IsFull: false
* IsEmpty: false

**LAB ACTIVITY (STACK)**

import java.util.Stack;

public class SimpleStackExample {

public static void main(String[] args) {

        Stack<Character> stack = new Stack<>(); // Create a stack

stack.push('A'); // Push A

stack.push('L'); // Push L

stack.push('I'); // Push I

stack.push('C'); // Push C

stack.push('E'); // Push E

 stack.pop(); // Removes E

        System.out.println("Top: " + stack.peek()); // Top (C)

stack.pop(); // Removes C

 stack.pop(); // Removes I

        System.out.println("Top: " + stack.peek()); // Top (L)

stack.push('O'); // Push O

stack.push('N'); // Push N

        System.out.println("Top: " + stack.peek());

        System.out.println("Size: " + stack.size());

        System.out.println("IsFull: false");

        System.out.print("Final Stack: ");

 for (char c : stack) {

            System.out.print(c + " ");

        }

    }}

**Final Output:**

Top: C

 Top: L

Top: N

 Size: 4

IsFull: false

Final Stack: A L O N

**LAB ACTIVITY (QUEUE)**

import java.util.LinkedList;

 import java.util.Queue;

 public class Main{

public static void main(String[] args) {

Queue<String> queue = new LinkedList<>();

queue.add("A");

queue.add("B");

queue.add("C");

queue.add("D");

queue.add("E");

System.out.println("Queue before dequeue: " + queue);

System.out.println("Front element before dequeue: " + queue.peek());

queue.remove();

queue.enqueue("K");

System.out.println("Front element after dequeue: " + queue.peek());

System.out.println("Queue after dequeue: " + queue);

}

}

**LABORATORY (QUEUE)**

import java.util.LinkedList;

import java.util.Queue;

public class Main {

 public static void main(String[] args) {

Queue<Character> queue = new LinkedList<>();

queue.add('J');

queue.add('A');

queue.add('P');

queue.add('A');

queue.add('N');

System.out.println("Front: " + queue.peek());

System.out.println("Rear: " +getRear(queue));

queue.remove();

queue.remove();

System.out.println("Front: " + queue.peek());

queue.add('I');

queue.add('G');

queue.remove();

System.out.println("Front: " + queue.peek());

System.out.println("Rear: " + getRear(queue));

queue.add('H');

queue.add('T');

queue.remove();

System.out.println("Front: " + queue.peek());

System.out.println("Rear: " + getRear(queue));

System.out.print("Final Queue: ");

for (char c : queue) {

 System.out.print(c + " ");

 }

}

**Final Output:**

Front: J

Rear: N

Front: P

Front: A

Rear: G

Front: N

Rear: T

 Final Queue: N I G H T