**1. Difference Between a Queue and a Stack**

* **Queue**:
  + Follows the **First In, First Out (FIFO)** principle.
  + Elements are added at the end (enqueue) and removed from the front (dequeue).
  + Example: A line of people waiting to buy tickets. The first person in line is served first.
* **Stack**:
  + Follows the **Last In, First Out (LIFO)** principle.
  + Elements are added to the top (push) and removed from the top (pop).
  + Example: A stack of plates. The last plate added on top is the first one to be removed.

**2. User Implementation of a Stack Using an ArrayList**

**(a),(b)**

**public class GenericStackException extends RuntimeException {**

**public GenericStackException(String message) {**

**super(message);**

**}**

**}**

**(c)**

**import java.util.ArrayList;**

**public class GenericStack<T> {**

**private ArrayList<T> items;**

**private int top;**

**public GenericStack() {**

**items = new ArrayList<>();**

**top = 0;**

**}**

**private boolean isEmpty() {**

**return top == 0;**

**}**

**public void push(T item) {**

**items.add(item);**

**top++;**

**}**

**public T pop() {**

**if (isEmpty()) {**

**throw new GenericStackException("Underflow Error");**

**}**

**return items.remove(--top);**

**}**

**}**

**(d)**

**public class StackDriver {**

**public static void main(String[] args) {**

**GenericStack<Integer> stack = new GenericStack<>();**

**stack.push(1);**

**stack.push(2);**

**stack.push(3);**

**stack.push(4);**

**try {**

**System.out.println(stack.pop());**

**System.out.println(stack.pop());**

**System.out.println(stack.pop());**

**System.out.println(stack.pop());**

**System.out.println(stack.pop());**

**} catch (GenericStackException e) {**

**System.out.println(e.getMessage());**

**}**

**}**

**}**

**3.**

**LinkedList<String> list = new LinkedList<>();**

**list.addFirst("First Node");**

**list.addLast("Last Node");**

**list.add("Another Last Node");**

**4.**

**public class Course implements Comparable<Course> {**

**private String code;**

**private String name;**

**public int compareTo(Course other) {**

**return this.code.compareTo(other.code);**

**}**

**}**

**5.**

**import java.util.HashMap;**

**import java.util.Map;**

**public class CourseMap {**

**public static void main(String[] args) {**

**Map<String, String> courses = new HashMap<>();**

**courses.put("CIT", "Computing and Information Technology");**

**courses.put("CHI", "Childcare and Early Education");**

**courses.put("MVS", "Motor Vehicle Systems");**

**courses.put("BTH", "Beauty Therapy");**

**courses.put("GDE", "Graphic Design");**

**for (Map.Entry<String, String> entry : courses.entrySet()) {**

**System.out.println(entry.getKey() + ": " + entry.getValue());**

**}**

**String courseName = courses.get("CIT");**

**System.out.println("Course Name for CIT: " + courseName);**

**}**

**}**