Queues and stacks are fundamental data structures that are widely used in many programming scenarios. In this tutorial, we'll explore how they work in Java and solve three basic problems using each data structure.

1. Stack

A **stack** is a **Last-In, First-Out (LIFO)** data structure. This means that the last element added to the stack is the first one to be removed. Operations on a stack typically include:

- **push()**: Add an element to the top of the stack.
- **pop()**: Remove the element from the top of the stack.
- **peek()**: View the element at the top of the stack without removing it.
- **isEmpty()**: Check if the stack is empty.

2. Queue

A **queue** is a **First-In, First-Out (FIFO)** data structure. The first element added to the queue is the first one to be removed. Queue operations include:

- **enqueue()**: Add an element to the back of the queue.
- **dequeue()**: Remove an element from the front of the queue.
- **peek()**: View the element at the front without removing it.
- **isEmpty()**: Check if the queue is empty.