

## Stacks

- Last In, First Out (LIFO) data structure
  - The last piece of data put onto the stack is the first one removed.
  - The last item is on the “top” of the stack.
  - Analogous to a stack of plates
- Stack Methods
  - `Stack<Object> s = new Stack<Object>();`
  - `push(Object obj)` : puts an item on the top of the stack
  - `pop()` : returns the item on the top of the stack and removes it
  - `peek()` : returns the item on the top of the stack without removing it
- Under the hood, the Java Stack class uses an ArrayList to store items. This is efficient because items are always removed from the end of the list.

## Queues

- First In, First Out (FIFO) data structure
  - The first piece of data put into the queue is the first one removed.
  - The last item is on the “tail” of the queue and the first item is at the “head” of the queue.
  - Analogous to a line of people
- Queue Methods
  - `Queue<Object> s = new LinkedList<Object>();`
    - `LinkedList` implements the `Queue` interface
  - `add(Object obj)` : adds an item to the tail of the queue
  - `remove()` : returns the next item in the queue and removes it
  - `peek()` : returns the next item in the queue without removing it
- A Queue is implemented as a LinkedList because the first item in the list is always removed.