1.

The FIFO principle governs linear data structures in queues. The front, or head, has items withdrawn, while the back, or tail, adds items. This data structure can be represented by a collection of items with existing elements removed and new ones added at the other end.

2.

Front (Head) [element1] [element2] [element3] ... [elementN] Rear (Tail)

3.

- First-In-First-Out (FIFO) behavior: Elements are processed in the order they were added to the queue.

- Enqueue: Adding an element to the rear/tail of the queue.

- Dequeue: Removing an element from the front/head of the queue.

- Peek/Front: Viewing the element at the front/head without removing it.

- is Empty: Checking if the queue is empty.

- Size: Determining the number of elements in the queue.

4.

FIFO is an acronym for "First-In-First-Out" (FIFO), which explains queuing. It involves the first item in a queue being withdrawn, and components are removed from the front and added to the back. As elements are enqueued, they move farther, and as removed, existing elements move forward. The oldest items are processed first.

5.

The three instances when a queue could be empty are:

- Initially Empty: When a queue is created, it is initially empty and devoid of any components.

- After Dequeuing All Elements: The queue will finally empty itself if all the elements are dequeued.

- Following Initialization and Clearing: A queue will become empty once more following initialization and clearing of all its components.

6.

Queues are commonly found in real-life situations and are used to manage and process tasks in a fair and orderly manner. Here are some instances:

- Supermarket Checkout: Customers waiting in line to pay for their items form a queue. The first customer in line is served first.

- Print Queue: Print jobs sent to a printer are managed in a queue, and the printer processes them in order for them to arrive.

- Traffic at Toll Booths: Vehicles waiting in line at a toll booth create a queue. The first car to arrive will be the first to pass through the booth.

- Waiting for a Bus: People waiting at a bus stop form a queue, and the first person to arrive will be the first to board the bus.

- Handling Requests in a Web Server: When multiple requests are made to a web server, they are placed in a queue and processed one by one in the order they were received.

-Restaurant Waiting List: In busy restaurants, customers waiting for a table are placed in a queue, and the next available table is assigned to the first group in the queue.

These examples demonstrate the practical applications of queues in various everyday scenarios, ensuring efficient and organized task handling.