Objective of the Circular Queue

The primary goal of implementing a Circular Queue is to provide an efficient way to manage a fixed-size array for storing elements while allowing for dynamic insertion and deletion. Unlike a linear queue, which can become inefficient as elements are removed (leading to unused space at the front), a circular queue wraps around when it reaches the end of the array. This allows for better utilization of space and prevents wastage.

Key Features

1. **Fixed Size**: The Circular Queue has a predefined maximum size, which limits how many elements it can hold.

2. **Circular Nature**: When the end of the array is reached, new elements can be added at the beginning of the array if there is space available.

3. **Efficient Operations**:

- **Enqueue (Insertion)**: Adds an element to the rear of the queue.
- **Dequeue (Deletion)**: Removes an element from the front of the queue.
- Both operations operate in O(1) time complexity.

4. **Tracking Indices**: It maintains two pointers (or indices) to track the front and rear of the queue, along with a count of the current number of elements.