```java
class Queue {

        private int front; // front represents the index position of the first element in the queue

        private int rear; // rear represents the index position of the last element in the queue

        private int maxSize; // maxSize represent the maximum number of elements that can be
stored in the queue

        private String arr[];


        Queue(int maxSize) {

                this.front = 0; // front is 0 when the queue is created

                this.rear = -1; // rear is -1 when the queue is created

                this.maxSize = maxSize;

                this.arr = new String[maxSize];

        }


        // Checking if the queue is full or not

        public boolean isFull() {

                if (rear == maxSize - 1) {

                        return true;

                }

                return false;

        }


        // Adding a new element to the rear of queue

        public boolean enqueue(String data) {

                if (isFull()) {

                        return false;

                } else {

                        arr[++rear] = data;

                        return true;

                }
```

```java
        }


        // Displaying all the elements in the queue
        public void display() {
                if (isEmpty())
                        System.out.println("Queue is empty!");
                else {
                        System.out.println("Displaying queue elements");
                        for (int index = front; index <= rear; index++) {
                                System.out.println(arr[index]);
                        }
                }
        }


        // Checking if the queue is empty or not
        public boolean isEmpty() {
                if (front > rear)
                        return true;
                return false;
        }


        // Removing an element from the front of queue
        public String dequeue() {
                if (isEmpty()) {
                        return "empty";
                } else {
                        String data = arr[this.front];
                        arr[front++] = null;
                        return data;
                }
        }
```

```java
        }

class Tester {

        public static void main(String[] args) {

                Queue queue = new Queue(5);
                System.out.println("Queue created.\n");


                if (queue.enqueue("Joe"))
                        System.out.println("The element is enqueued to the queue!\n");
                else
                        System.out.println("Queue is full!\n");


                if (queue.enqueue("Jack"))
                        System.out.println("The element is enqueued to the queue!\n");
                else
                        System.out.println("Queue is full!\n");


                if (queue.enqueue("Eva"))
                        System.out.println("The element is enqueued to the queue!\n");
                else
                        System.out.println("Queue is full!\n");


                if (queue.enqueue("Mia"))
                        System.out.println("The element is enqueued to the queue!\n");
                else
                        System.out.println("Queue is full!\n");


                if (queue.enqueue("Luke"))
                        System.out.println("The element is enqueued to the queue!\n");
```

```java
			else
					System.out.println("Queue is full!\n");


			queue.display();


			if (queue.enqueue("Emma"))
					System.out.println("The element is enqueued to the queue!\n");
			else
					System.out.println("Queue is full!\n");


			String dequeuedElement = queue.dequeue();
			if (dequeuedElement == "empty")
					System.out.println("Queue is empty\n");
			else
					System.out.println("The element dequeued is : " + dequeuedElement +
"\n");


			dequeuedElement = queue.dequeue();
			if (dequeuedElement == "empty")
					System.out.println("Queue is empty\n");
			else
					System.out.println("The element dequeued is : " + dequeuedElement +
"\n");


			dequeuedElement = queue.dequeue();
			if (dequeuedElement == "empty")
					System.out.println("Queue is empty\n");
			else
					System.out.println("The element dequeued is : " + dequeuedElement +
"\n");


			dequeuedElement = queue.dequeue();
```

```java
            if (dequeuedElement == "empty")

                    System.out.println("Queue is empty\n");

            else

                    System.out.println("The element dequeued is : " + dequeuedElement +
"\n");


            dequeuedElement = queue.dequeue();

            if (dequeuedElement == "empty")

                    System.out.println("Queue is empty\n");

            else

                    System.out.println("The element dequeued is : " + dequeuedElement +
"\n");


            dequeuedElement = queue.dequeue();

            if (dequeuedElement == "empty")

                    System.out.println("Queue is empty\n");

            else

                    System.out.println("The element dequeued is : " + dequeuedElement +
"\n");
        }
}
```

Output:

Queue created.

The element is enqueued to the queue!

The element is enqueued to the queue!

The element is enqueued to the queue!

The element is enqueued to the queue!

The element is enqueued to the queue!

Displaying queue elements

Joe

Jack

Eva

Mia

Luke

Queue is full!

The element dequeued is : Joe

The element dequeued is : Jack

The element dequeued is : Eva

The element dequeued is : Mia

The element dequeued is : Luke

Queue is empty