




Tutorial 5

Student Name Kan Katpark **Student ID** 642115003

1. Develop a simple queue application to support CAMT student helpdesk service operation. The system shall be able to determine the number of queue the last student has to wait in the queue. The first student in the queue is assumed getting serve by the CAMT helpdesk staff.

Hint: you can modify the given code

scenarios	Output
	The queue is empty
	queue size= 1
	queue size= 5

2. Write a method `size()` to determine a number of waiting queue

```
public int size() {  
    return this.size;  
}
```

3. Write a method `isEmpty()` to determine whether the queue is empty or not

```
public boolean isEmpty() {  
    return !(size > 0);  
}
```

4. Write a method `isFull()` to determine whether the queue capacity is full or not

```
public boolean isFull() {  
    return this.size == this.maxQueue ;  
}
```

5. Implement the enqueue and dequeue

```
public void enqueue(Object x) {  
    if (this.size >= this.maxQueue) {  
        System.out.println("Queue is full! ");  
        return ;  
    }  
  
    size++;  
    collection.addElement(x);  
}  
  
public Object dequeue() {  
    if (this.size <= 0) {  
        System.out.println("Queue is empty! ");  
        return null;  
    }  
  
    size--;  
    return this.collection.remove(0);  
}
```

6. Show how the methods are used in the main()

```
final int MAX_QUEUE_SIZE = 10;
Queue myqueue = new Queue(MAX_QUEUE_SIZE);

Scanner scan = new Scanner(System.in);
String input = "";

while (!(input.equals("quit"))) {
    System.out.print("Please Enter student
name or 'quit' to exit program: ");
    input = scan.next();

    if (input.equals("quit")) {
        break;
    }

    if (myqueue.size() >= MAX_QUEUE_SIZE) {
        System.out.println("Dequeue: " +
myqueue.dequeue());
        myqueue.printQueue();
    }
    myqueue.enqueue(input);
    System.out.println("Enqueue: " + input);
    myqueue.printQueue();
}
```

7. Submit your Java source code consisting the main class and your queue class that produce the following output (3 points)

MyQueue.java (main class)

```
import java.util.*;

public class MyQueue {

    public static void main(String[] args) {

        final int MAX_QUEUE_SIZE = 5;

        Queue myqueue = new Queue(MAX_QUEUE_SIZE);

        Scanner scan = new Scanner(System.in);
        String input = "";

        while (!(input.equals("quit"))) {

            System.out.print("Please Enter student name or 'quit' to exit
program: ");

            input = scan.next();

            if (input.equals("quit")) {

                break;

            }

            if (myqueue.size() >= MAX_QUEUE_SIZE) {

                System.out.println("Dequeue: " + myqueue.dequeue());
                myqueue.printQueue();

            }

            myqueue.enqueue(input);
            System.out.println("Enqueue: " + input);
            myqueue.printQueue();

        }

    }

}
```

Queue.java (queue class)

```
import java.util.Vector;

class Queue {
    private Vector<Object> collection;
    private int maxQueue;
    private int size;
    public Queue(int max) {
        this.maxQueue = max;
        this.collection = new Vector<Object>();
        this.size = 0;
    }
    public void enqueue(Object x) {
        if (this.size >= this.maxQueue) {
            System.out.println("Queue is full! ");
            return;
        }
        size++;
        collection.addElement(x);
    }

    public Object dequeue() {
        if (this.size <= 0) {
            System.out.println("Queue is empty! ");
            return null;
        }
        size--;
        return this.collection.remove(0);
    }

    public boolean isEmpty() {
        return !(size > 0);
    }
}
```

```

    public int front() {
        return 0;
    }

    public int tail() {
        return this.size - 1;
    }

    public int size() {
        return this.size;
    }

    public void printQueue() {
        String output = "[";

        for (int i = 0; i < maxQueue; i++) {
            try {
                output += collection.elementAt(i);
            } catch (ArrayIndexOutOfBoundsException e) {
                output += "NULL";
            }
            if (i != maxQueue - 1)
                output += ",";
        }
        output += "] first: " + front() + " rear: " + (tail()+1) + " Length: "
+ size();
        System.out.println(output);
    }

    public boolean isFull() {
        return this.size == this.maxQueue;
    }
}

```

Input:

Please enter student name (enter 'quite' for exit the program):

John

Man

Pree

Output:

Enqueue John

[John,null, null, null, null] [first = 0, rear = 1, length = 1]

Enqueue Man

[John,Man, null, null, null] [first = 0, rear = 2, length = 2]

Enqueue Pree

[John,Man,Pree, null, null] [first = 0, rear = 3, length = 3]

Dequeue John

[Man, Pree, null, null] [first = 1, rear = 2, length = 2]