



Assignment Queues

1 Objectives

1. Implement a queue using linked representation.
2. Implement a queue using array representation.

2 Queue Implementation

Create a queue class to implement the following **IQueue** interface

```
public interface IQueue {  
    /**  
    * Inserts an item at the queue front.  
    */  
    public void enqueue(Object item);  
    /**  
    * Removes the object at the queue rear and returns it.  
    */  
    public Object dequeue();  
    /**  
    * Tests if this queue is empty.  
    */  
    public boolean isEmpty();  
    /**  
    * Returns the number of elements in the queue  
    */  
    public int size();  
}
```



Your class should inherit from this interface twice; once to implement a queue using linked-based representation, and once using an array based implementation and supply all it's method with the exact signature.

In the array based implementation, your queue won't have more than n elements where n is a parameter in your class constructor and a user defined input in your testing class.

You should provide any JUnit tests for testing both the implementation of the Queue.

In order to distinguish between the two queue implementations: array-based and linked-based, you need to implement the following two empty interfaces, respectively.

```
public interface ILinkedBased { }
```

```
public interface IArrayBased { }
```

3 Deliverables

- This assignment will be delivered alongside the next assignment.
- You should work in teams of up to 3 members.
- You should use your own data structures which were implemented in the previous assignments. Don't use any built-in data structure.
- Late submission is accepted for only one week.
- Delivering a copy will be severely penalized for both parties, so delivering nothing is so much better than delivering a copy.