# Examination 2 (Java) : Implement This Skeleton Code

***Java only**

```
package jp.co.worksap.global;

import java.util.NoSuchElementException;

/**
 * The Queue class represents an immutable first-in-first-out (FIFO) queue of objects.
 * @param <E>
 */
public class ImmutableQueue<E> {
	/**
	 * requires default constructor.
	 */
	public ImmutableQueue() {
		// modify this constructor if necessary, but do not remove default constructor
	}

	// add other constructors if necessary

	/**
	 * Returns the queue that adds an item into the tail of this queue without modifying this queue.
	 * <pre>
	 * e.g.
	 * When this queue represents the queue (2, 1, 2, 2, 6) and we enqueue the value 4 into this queue,
	 * this method returns a new queue (2, 1, 2, 2, 6, 4)
	 * and this object still represents the queue (2, 1, 2, 2, 6) .
	 * </pre>
	 * If the element e is null, throws IllegalArgumentException.
	 * @param e
	 * @return
	 * @throws IllegalArgumentException
	 */
	public ImmutableQueue<E> enqueue(E e) {
		return null;
	}

	/**
	 * Returns the queue that removes the object at the head of this queue without modifying this queue.
	 * <pre>
	 * e.g.
	 * When this queue represents the queue (7, 1, 3, 3, 5, 1) ,
	 * this method returns a new queue (1, 3, 3, 5, 1)
	 * and this object still represents the queue (7, 1, 3, 3, 5, 1) .
	 * </pre>
	 * If this queue is empty, throws java.util.NoSuchElementException.
	 * @return
	 * @throws java.util.NoSuchElementException
	 */
	public ImmutableQueue<E> dequeue() {
		return null;
	}

	/**
	 * Looks at the object which is the head of this queue without removing it from the queue.
	 * <pre>
	 * e.g.
	 * When this queue represents the queue (7, 1, 3, 3, 5, 1),
	 * this method returns 7 and this object still represents the queue (7, 1, 3, 3, 5, 1)
	 * </pre>
	 * If the queue is empty, throws java.util.NoSuchElementException.
	 * @return
	 * @throws java.util.NoSuchElementException
	 */
	public E peek() {
		return null;
	}

	/**
	 * Returns the number of objects in this queue.
	 * @return
	 */
	public int size() {
		return -1;
	}
}
```