420-B31

# Lab 9 – The Queue ADT Answers

# Part A - Question 1

1. Fill in the blank entries for the following table. Remember that the object state for a queue is the front, rear, size and the contents of the queue itself.

| Method | Object State | Return value |
| --- | --- | --- |
| Queue <String> q =  new LinkedQueue<String> (); | Front = null  Rear = null  Size = 0 | Returns a LinkedQueue |
| q.enqueue("C"); | Front = “C”  Rear = The 2nd queue slot  Size = 1  C |  |
| q.enqueue("A"); | Front = “C”  Rear = The slot after “A”  Size = 2  C, A |  |
| q.enqueue("M"); | Front = “C”  Rear = The slot after “M”  Size = 3  “C” A M |  |
| q.peek(); |  | Returns C |
| q.dequeue(); | Front = “A”  Rear = The slot after “M”  Size = 2  A M | Returns C |
| q.peek(); |  | Returns A |
| q.isEmpty(); |  | False |
| q.dequeue(); | Front = M  Rear = The slot after M  Size = 2  M | Returns A |
| q.dequeue(); | Front = null  Rear = null  Size = 0 | Returns M |
| q.peek(); |  | EmptyQueueException |
| q.dequeue(); |  | EmptyQueueException |

# Part B - Priority Queue using Adapter Design Pattern

## 1b. PriorityQueue operations mapped to Queue operations

|  |  |
| --- | --- |
| PriorityQueue Operation | Queue Operation Equivalent |
| void enqueue(element) | Add (Size(E)) |
| void enqueue(priority, element) | Doesn’t exist |
| E dequeue() | Remove(0) |
| E peek() | Peek() |
| int size() | Size() |
| boolean isEmpty() | isEmpty() |
| void clear() | Clear() |