```java
import org.junit.jupiter.api.*;
import static org.junit.jupiter.api .Assertions.*;
import java.util.NoSuchElementException;
public class QueueTest {
    Queue queue;
    @BeforeEach
    void setUp(){
        queue = new Queue();
    }
    @AfterEach
    void breakDown() { queue = null;}
    @Test
    @DisplayName("Testing enqueueing without error, Failure and Faults" )
    void testEnQ(){
        queue.enq(10);
        queue.enq(25);
        queue.enq(105);
        queue.enq(20);
        assertEquals(4,queue.len());
        assertFalse(queue.Empty());
    }
    @Test
    @DisplayName("Testing dequeueing without error, Failure and Faults" )
    void testDeQ(){
        queue.enq(10);
        queue.enq(25);
        queue.enq(105);
        queue.enq(20);
        assertEquals(10,queue.deq());
        assertEquals(25,queue.deq());
        assertEquals(105,queue.deq());
        assertEquals(20,queue.deq());
        assertTrue(queue.Empty());
    }
    @Test
    @DisplayName("Testing size of queue without error ,Failure and Faults" )
    void testL(){
        assertEquals(0,queue.len());
```

```
39          queue.enq(5);
40          queue.enq(4);
41          queue.enq(3);
42          assertEquals(3,queue.len());
43          queue.deq();
44          assertEquals(2,queue.len());
45          queue.clear();
46          assertEquals(0,queue.len());
47      }
48      @Test
49      @DisplayName("Testing an empty queue with Empty
    () and without error,Failure and Faults" )
50      void testEmpty(){
51          assertTrue(queue.Empty());
52          queue.enq(15);
53          assertFalse(queue.Empty());
54          queue.deq();
55          assertTrue(queue.Empty());
56      }
57      @Test
58      @DisplayName("Testing an empty queue with clear
    () and without error,Failure and Faults" )
59      void testClear(){
60          queue.enq(10);
61          queue.enq(25);
62          queue.enq(105);
63          queue.enq(20);
64          assertFalse(queue.Empty());
65          queue.clear();
66          assertEquals(0,queue.len());
67          assertTrue(queue.Empty());
68      }
69      @Test
70      @DisplayName("Enqueueing Null value to perform
    Failure" )
71      void enqNullValue(){
72          queue.enq(null);
73          assertFalse(queue.Empty());
74  //          assertTrue(queue.Empty());// Failed test
    case cuz we are trying to push a null value
75      }
```

```
 76     @Test
 77     @DisplayName("Testing the size of empty queue
    with clear() and without error,Failure and Faults" )
 78     public void sizeCheckAfterEmpty(){
 79         queue.enq(10);
 80         queue.enq(25);
 81         queue.enq(105);
 82         queue.enq(20);
 83 //         queue.Empty(); this will just empty the
    queue and still the space created for 4 variables
    will still be there reserved
 84         queue.clear();
 85         assertEquals(0,queue.len());
 86     }
 87     @Test
 88     public void testCheckNull(){
 89         assertEquals(0,queue.len());
 90         queue.enq(null);
 91         assertNull(queue.check());
 92     }
 93     @Test
 94     public void checkEmptyOnEmptyQueue(){
 95         assertEquals(0,queue.len());
 96         assertTrue(queue.Empty());
 97     }
 98     @Test
 99     @DisplayName("Dequeueing null value form queue
    to perform Error" )
100     public void deqNullItem(){
101         queue.enq(null);
102         assertNull(queue.deq());
103     }
104     @Test
105     @DisplayName("Enqueueing a large number of
    elements to test capacity limits")
106     void testEnqLarge() {
107         for (int i = 0; i < 100000; i++) {
108             queue.enq(i);
109         }
110         assertEquals(100000, queue.len());
111         assertFalse(queue.Empty());
```

```
112         queue.clear();
113         assertTrue(queue.Empty());
114     }
115     @Test
116     @DisplayName("Dequeueing from an empty queue to
    perform Error")
117     void testDeqEmptyQueue() {
118         assertThrows(NoSuchElementException.class
    , () -> {
119             queue.deq(); // This should throw an
    error when dequeuing from an empty queue
120         });
121     }
122     @Test
123     @DisplayName("Null enqueue followed by valid
    operations to detect Failure")
124     void testNullEnqWithValidOperations() {
125         assertThrows(IllegalArgumentException.class
    , () -> {
126             queue.enq(null); // This should fail due
     to the null check
127         });
128         queue.enq(1);
129         queue.enq(2);
130         assertEquals(1, queue.deq());
131         assertEquals(1, queue.len());
132     }
133     @Test
134     @DisplayName("Clear queue and perform operations
     to find Faults")
135     void testClearAndOperate() {
136         queue.enq(10);
137         queue.enq(20);
138         queue.clear(); // Clear the queue
139         assertTrue(queue.Empty());
140         assertThrows(NoSuchElementException.class
    , () -> {
141             queue.deq(); // Attempting to dequeue
    after clearing should cause an error
142         });
143     }
```

```
144     @Test
145     @DisplayName("Checking the state after mixed
    enqueue and dequeue operations")
146     void testMixedOperations() {
147         queue.enq(5);
148         queue.enq(10);
149         queue.deq(); // Remove 5
150         queue.enq(15);
151         queue.deq(); // Remove 10
152         queue.enq(20);
153         assertEquals(15, queue.deq()); // Should
    dequeue 15
154         assertEquals(20, queue.deq()); // Should
    dequeue 20
155         assertTrue(queue.Empty());
156     }
157     @Test
158     @DisplayName("Check consistency with alternate
    enqueue and dequeue")
159     void testAlternateEnqDeq() {
160         queue.enq(1);
161         assertEquals(1, queue.deq());
162         queue.enq(2);
163         queue.enq(3);
164         assertEquals(2, queue.deq());
165         queue.enq(4);
166         assertEquals(3, queue.deq());
167         assertEquals(4, queue.deq());
168         assertTrue(queue.Empty());
169     }
170     @Test
171     @DisplayName("Testing duplicate values in the
    queue")
172     void testDuplicateValues() {
173         queue.enq(5);
174         queue.enq(5);
175         queue.enq(5);
176         assertEquals(3, queue.len());
177         assertEquals(5, queue.deq());
178         assertEquals(5, queue.deq());
179         assertEquals(5, queue.deq());
```

```
180         assertTrue(queue.Empty());
181     }
182     @Test
183     @DisplayName("Testing state after multiple
    clears")
184     void testMultipleClears() {
185         queue.enq(1);
186         queue.enq(2);
187         queue.clear(); // First clear
188         assertTrue(queue.Empty());
189         queue.clear(); // Second clear should not
    break anything
190         assertEquals(0, queue.len());
191     }
192     @Test
193     @DisplayName("Peeking from an empty queue")
194     void testPeekEmptyQueue() {
195         assertNull(queue.check());
196         assertTrue(queue.Empty());
197     }
198     @Test
199     @DisplayName("Mix of null and valid objects in
    queue")
200     void testNullAndValidMix() {
201         queue.enq(null);
202         queue.enq(42);
203         assertNull(queue.deq()); // Null value
    dequeued first
204         assertEquals(42, queue.deq());
205         assertTrue(queue.Empty());
206     }
207 }
```