Derive input space partitioning tests for the BoundedQueue class, which was introduced in STC03,

with the following signature:

public BoundedQueue (int capacity);

public void Enqueue (Object X);

public Object Dequeue ();

public boolean IsEmpty ();

public boolean IsFull ();

Assume the usual semantics for a queue with a fixed, maximal capacity. Try to keep your partitioning

simple–choose a small number of partitions and blocks.

Identify all of the variables. Don’t forget the state variables.

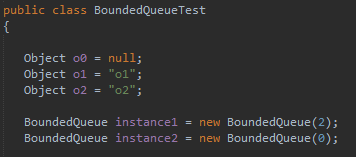
Identify several characteristics that suggest partitions.

Identify the blocks in the partition for each characteristic. Designate one block in each partition

as the “Base” block.

Define values for the blocks.

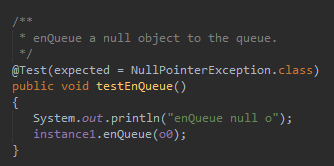
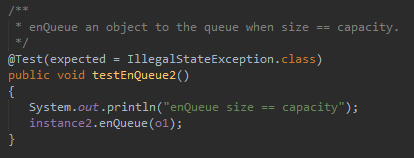
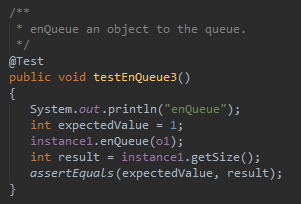
Define a test set that satisfies base choice coverage (BCC).



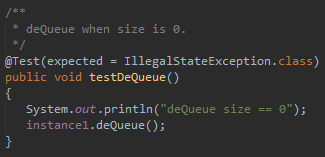
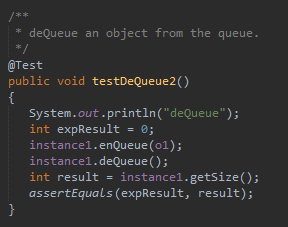
-public BoundedQueue (int capacity);

capacity is the only variable that causes different characteristics. capacity less than 0 causes the object to not be created and an exception to be thrown. -1 is an example of this block. Zero and positive values are another block. 1 is an example of this block. We can have two tests with the mentioned values to cover all the partitions.

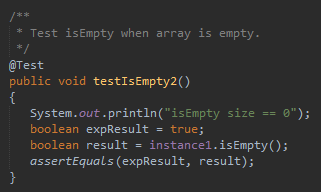
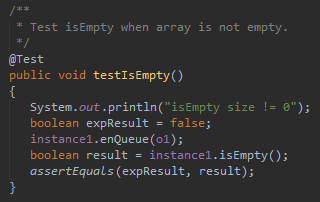
-public void Enqueue (Object X);

The object being enqueued, size of the queue, and the capacity of the queue are the variables that cause different characteristics. If we try to enqueue a null object, an exception will be thrown.  
  
  
  
If the object is not null and we try to enqueue when the size is equal to capacity, another exception will by thrown.   
  
  
  
If the object is not null and we try to enqueue to a queue that is not full, we get another characteristic.  
  


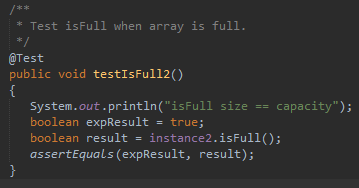
-public Object Dequeue ();

Size of the queue is the variable that cause different characteristics. If we try to dequeue when the size is 0, we will get an exception.   
  
  
  
If we try to dequeue when the size is not 0, we will get another characteristic.  
  


-public boolean IsEmpty ();

Size is the variable that causes different characteristics. If we call it when the size is 0, we will get true.  
  
  
  
If we call it when the size is not 0, we will get false.  
  


-public boolean IsFull ();

Size and capacity are the variables that cause different characteristics. If we call it when the size equals capacity, we will get true.  
  
  
  
If we call it when the size is not equal to capacity, we will get false.  
  
