

(a) List all the input variables, including the state variables.							
input variables:	int capacity	Object o					
state variables:	int capacity	int size, front, back	Object[] elements				
(b) Define the characteristics of the input variables. Make sure you cover all input variables.							
Method	Params	Returns	Values	Exception	Ch ID	Characteristic	Covered by
BoundedQueue	int				C1	Constructor	
				IllegalArgumentException	C2	If argument is less than 0	
enqueue	Object				C3	Make o the newest element of the queue	
				NullPointerException	C4	If argument is null	
				IllegalStateException			C7
deQueue	state	Object	Object o		C5	Remove and return oldest element of the queue	
				IllegalStateException			C6
isEmpty	state	boolean	true, false		C6	If queue is empty	
isFull	state	boolean	true, false		C7	If queue is full	
(c) Partition the characteristics into blocks. Designate one block in each partition as the "Base" block.							
ID	Characteristic	BoundedQueue (int capacity)	enqueue (Object o)	deQueue ()	isEmpty()	isFull()	
C1	Constructor	O					
C2	If argument is less than 0	O					
C3	Make o the newest element of the queue		O				
C4	If argument is null		O				
C5	Remove and return oldest element of the queue			O			
C6	If queue is empty			O	O		
C7	If queue is full		O			O	
(d) Define values for each block.							
ID	Partition	Base					
C1	Constructor						
C2	{true, false}	F					
C3	{true, false}	T					
C4	{true, false}	F					
C5	{true, false}	T					
C6	{true, false}	F					
C7	{true, false}	F					
(e) Define a test set that satisfies Base Chose Coverage (BCC). Write your tests with the values from the previous step. Be sure to include the test oracles.							
Method	Characteristics	Test Requirements	Infeasible TRs	Revised TRs	# TRs		
BoundedQueue	C1, C2	{TF, FF, TT}	FF, TT	FF/TT -> FT	2		
enqueue	C3, C4, C7	{TFF, FFF, TTF, TFT}	FFF, TTF, TFT	TTF -> FTF TFT -> FFT	3		
deQueue	C5, C6	{TF, FF, TT}	FF, TT	FF/TT -> FT	2		
isEmpty	C6	{F, T}	none	n/a	2		
isFull	C7	{F, T}	none	n/a	2		