HW2　310551019_王璽禎						
(a) List all of the input variables, including the state variables.						
Input variables						
State variables	BoundedQueue	enQueue	deQueue	isEmpty	isFull	
(b) Define the characteristics of the input variables. Make sure you cover all input variables.						
Method	**Params**	**Returns**	**Values**	**Exception**	**Ch ID**	**Characteristic**
BoundedQueue	state				C1	Constructor
				IllegalArgumentException	C2	If argument is less than 0
enQueue	state				C3	Make o the newest element of the queue
				NullPointerException	C4	If argument is null
				IllegalStateException		
deQueue	state	object	o		C5	Remove and return oldest element of the queue
				IllegalStateException		
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	V				
C2	If argument is less than 0	V				
C3	Make o the newest element of the queue		V			
C4	If argument is null		V			

C5	Remove and return oldest element of the queue			V		
C6	If queue is empty			V	V	
C7	If queue is full		V			V
(d) Define values for each block.						
C1	Constructor	V				
C2	If argument is less than 0	V				
C3	Make o the newest element of the queue		V			
C4	If argument is null		V			
C5	Remove and return oldest element of the queue			V		
C6	If queue is empty			V	V	
C7	If queue is full		V			V
(e) Define a test set that satisfies Base Choice 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, TT, FT}	{FT}	FT->FF	3	
enQueue	C3, C4, C7	{TFF, TTF, TFT, FTT}	{FTT}	FTT->FFF	4	
deQueue	C5, C6	{TF, TT, FT}	{TT}	TT->FF	3	
isEmpty	C6	{T, F}	none	n/a	2	
isFull	C7	{T, F}	none	n/a	2	