Testing Type Details: Unit Testing
Tester Name: Group1
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
1	Enter priority number 0			**1-call Operation constructor with priority number 0**	1- initialize operation constructor	Throw illegal argument exception	exit	Throw illegal argument exception	Y
2	Enter priority number -1			**1-call Operation constructor with priority number -1**	1- initialize operation constructor	Throw illegal argument exception	exit	Throw illegal argument exception	Y
3	Enter priority number 1			**1-call Operation constructor with priority number 1**	1- initialize operation constructor 2- Call getpriority()	work normally	exit	Work normally	Y
4	Enter priority number 10			**1-call Operation constructor with priority number 10**	1- initialize operation constructor 2- Call getpriority()	Work normally	exit	Work normally	Y
5	Enter priority number 100			**1-call Operation constructor with priority number 100**	1- initialize operation constructor	Throw illegal argument exception	exit	Throw illegal argument exception	Y
6	Enter priority number 9	b		**1-call Operation constructor with priority number 9**	1- initialize operation constructor 2- Call getpriority()	Work normally	exit	worknormally	Y

Testing Type Details: Unit Testing
Tester Name: Group1
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
7	Enter priority number 5			**1-Intailze Operation constructor with priority number 5**	1- initialize operation constructor 2- Call getpriority()	Work normally	exit	Work normally	Y
8	Enter priority number 10 and increment it			**1-Inalize Operation constructor with priority number 10 2-incrementprioity(20)**	1- initialize operation constructor 2- Call incrementPrioity(20)	Throws illegal argument exception	exit	Throw illegal argument exception	Y
9	Enter ID negative number			**1-Inalize Operation constructor with ID number -1**	1- initialize operation constructor with Test data	Throws illegal argument exception	exit	Doesn't throw anything and crash	Y
10	Enter arrival Time negative number			**1-Inalize Operation constructor with arrivalTime number -2**	1- initialize operation constructor with Test Data	Throws illegal argument exception	exit	Throw illegal argument exception	Y
11	Enter exeTime negative number	c		**1-Inalize Operation constructor with exeTime -1**	1- initialize operation constructor with Test Data	Throws illegal argument exception	exit	Throw illegal argument exception	Y

Testing Type Details: Unit Testing
Tester Name: Group1
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
12	Enter exeTime zero			**1-Inalize Operation constructor with exeTime 0**	1- initialize operation constructor with Test Data	Throws illegal argument exception	exit	Throw illegal argument exception	Y
13	Enter valid data of the constructor			**1-Inalize Operation constructor with all valid data**	1- initialize operation constructor with Test Data	Work normally	exit	Work normally	Y
14	Enter exetime 1 and decrement it two times with decrementTimeLeft(2)			**1-Inalize Operation constructor with exeTime 1** **2- decrementTimeleft(2)**	1- initialize operation constructor with Test Data	Throws illegal argument exception	exit	Throw illegal argument exception	Y
15	Enter a valid constructor and call getWaiting( ) as Response time is			**1-Inalize Operation constructor with valid input**	1- initialize operation constructor with Test Data 2- call getwaiting()	Throws illegal argument exception	exit	Throw illegal argument exception	Y

Testing Type Details: Unit Testing
Tester Name: Group1
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
	MAX_VALUE so it return -1 in getTAT() but getwait will be a negative number								
16	Enter a valid constructor And setResponse time with negative number			**1-Inalize Operation constructor with valid input 2-setResponseTime(-2)**	1- initialize operation constructor with Test Data 2- call setResponse(-2)	Throws illegal argument exception	exit	Throw illegal argument exception	Y
17	Enter a valid constructor And setResponse time with negative number And call getTAT will be negative number not -1			**1-Inalize Operation constructor with valid input 2-setResponseTime(-2)**	1- initialize operation constructor with Test Data 2- call getTATime()	Throws illegal argument exception	exit	Throw illegal argument exception	Y
18	FCFS enqueue it when the			**1-Inalize Operation constructor with arrival time less than timer**	**1-Inalize Operation constructor with arrival time 0 timer 2-Inalize 1 Operation with**	Throws illegal argument exception	exit	Throw illegal argument exception	Y

Testing Type Details: Unit Testing
Tester Name: Group1
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
	arrival time in the past and the timer has gone			**2-Inalize 1 Operations with valid input**	**valid input** **3-call enqueue()** **4-call consumeTimeUnit()** **5-call enqueue with the operation arrivalTime less than timer**				
19	if the queue in FCFSQ is null and call consumeTimeUnit			**1-intalize FCFSQ**	1- **intalize FCFSQ** 2- **call consumeTimeUnit()**	Return null	exit	Return null	Y
20	Put one operation in queue and consume time unit with exeTime 1			**1-intalize FCFSQ**	1- **intalize FCFSQ** 2- **inalize Operation with exeTime 1** 3- **call consumeTimeUnit()** 4- **call consumeTimeUnit() again**	Return null	exit	Return null	Y
21	Empty iterator in FCFS			**1-inalize FCFSQ**	1-inalize FCFSQ 2-getiterator()	Work normally	exit	Work normally	Y
22	SJFQ enqueue it when the arrival time in the past and the timer has gone			**1-Inalize Operation constructor with arrival time less than timer** **2-Inalize 1 Operations with valid input**	**1-Inalize Operation constructor with arrival time 0 timer** **2-Inalize 1 Operation with valid input** **3-call enqueue()** **4-call consumeTimeUnit()** **5-call enqueue with the operation arrivalTime less than timer**	Throws illegal argument exception	exit	Doesn't throw anything and crash	Y

Testing Type Details: Unit Testing
Tester Name: Group1
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
23	Temp queue is empty			1- **inalize SJFSQ**	1-inalize SJFSQ 2-inalize 3 operation with arrivalTime=Timer 3-enqueue them 4-call consumeTimeUnit()	Work normally and decrement the first Operation as it is lowest exetime	exit	Work normally	Y
24	In SJF queue is empty			**1-inalize SJFSQ**	1- Inalize SJFSQ 2- Call consumeTime() 3- Call consumeTime()again	Return null	Exit	Return null	Y
25	We will put all arrivalTime queues more than timer so it will be in Temp queues			1- **Inalize SJFSQ** 2- **Inalize 4 Operations with valid data but arrival time bigger than timer**	1- Inalize SJFSQ 2- Inalize 4 operations with valid data but arrival time bigger than timer 3- Enqueue it 4- Call 2 times ConsumeTime()	Return null in first call but second call work normally	exit	Return null in first call but second call work normally	Y
26	One process at queue			1- **inalize SJFSQ** 2- **inalize one operation with 1 exeTime**	1- inalize SJFSQ 2- inalize Operation with valid input but exeTime 1 3- enqueue it 4- ConsumeTimeUnit( ) 5- Call	Return null	exit	Return null	Y

Testing Type Details: Unit Testing
Tester Name: Group1
Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
					consumeTimeUnit( ) again				
27	In Preemptive SJFQ queue is empty			**1-inalize** PreemptiveSJFQ	1- Inalize PreemptiveSJFQ 2- Call consumeTime	Return null	exit	Return null	Y
28	We will put all arrivalTime queues more than timer so it will be in Temp queues			**1- Inalize** PreemptiveSJFQ **2- Inalize 4 Operations with valid data but arrival time bigger than timer**	1- Inalize PreemptiveSJFQ 2- Inalize 4 operations with valid data but arrival time bigger than timer 3- Enqueue it 4- Call consumeTimeUnit( ) two times	Return null in first call but second call work normally	exit	Return null in first call but second call work normally	Y
29	One process at queue			**1- inalize** Preemptive **SJFQ** **2- inalize one operation with 1 exeTime**	1- inalize SJFSQ 2- inalize Operation with valid input but exeTime 1 3- enqueue it 4- ConsumeTimeUnit( ) 5- Call consumeTimeUnit( ) again	Return null	exit	Return null	Y

Testing Type Details: Unit Testing

Tester Name: Group1

Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)

Testing Type Details: Unit Testing

Tester Name: Group1

Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)

Testing Type Details: Unit Testing

Tester Name: Group1

Test Environment Details

Scen #	Scenario Description	Req #	Cond #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
	112								