

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

Scen #	Scenario Description	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
1	Enter priority number 0			<u>1-call Operation constructor with priority number 0</u>	1- initialize operation constructor	Throw illegal argument exception	exit	Throw illegal argument exception	Y
2	Enter priority number -1			<u>1-call Operation constructor with priority number -1</u>	1- initialize operation constructor	Throw illegal argument exception	exit	Throw illegal argument exception	Y
3	Enter priority number 1			<u>1-call Operation constructor with priority number 1</u>	1- initialize operation constructor 2- Call getpriority()	work normally	exit	Work normally	Y
4	Enter priority number 10			<u>1-call Operation constructor with priority number 10</u>	1- initialize operation constructor 2- Call getpriority()	Work normally	exit	Work normally	Y
5	Enter priority number 100			<u>1-call Operation constructor with priority number 100</u>	1- initialize operation constructor	Throw illegal argument exception	exit	Throw illegal argument exception	Y
6	Enter priority number 9	b		<u>1-call Operation constructor with priority number 9</u>	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	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
7	Enter priority number 5			<u>1-Intailze Operation constructor with priority number 5</u>	1- initialize operation constructor 2- Call getpriority()	Work normally	exit	Work normally	Y
8	Enter priority number 10 and increment it			<u>1-Inalize Operation constructor with priority number 10</u> <u>2-incrementprioity(20)</u>	1- initialize operation constructor 2- Call incrementPrioity(20)	Throws illegal argument exception	exit	Throw illegal argument exception	Y
9	Enter ID negative number			<u>1-Inalize Operation constructor with ID number -1</u>	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			<u>1-Inalize Operation constructor with arrivalTime number -2</u>	1- initialize operation constructor with Test Data	Throws illegal argument exception	exit	Throw illegal argument exception	Y
11	Enter exeTime negative number	c		<u>1-Inalize Operation constructor with exeTime -1</u>	1- initialize operation constructor with Test	Throws illegal argument exception	exit	Throw illegal argument exception	Y

## Testing Type Details: Unit Testing

Tester Name: Group1

### Test Environment Details

Scen #	Scenario Description	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
					Data				
12	Enter exeTime zero			<u>1-Inalize Operation constructor with exeTime 0</u>	1- initialize operation constructor with Test Data	Throws illegal argument exception	exit	Throw illegal argument exception	Y
13	Enter valid data of the constructor			<u>1-Inalize Operation constructor with all valid data</u>	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)			<u>1-Inalize Operation constructor with exeTime 1</u> <u>2- decrementTimeleft(2)</u>	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 MAX_VALUE so it return -1 in getTAT() but getwait will be a negative number			<u>1-Inalize Operation constructor with valid input</u>	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	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
16	Enter a valid constructor And setResponse time with negative number			<u>1-Inalize Operation constructor with valid input</u> <u>2-setResponseTime(-2)</u>	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			<u>1-Inalize Operation constructor with valid input</u> <u>2-setResponseTime(-2)</u>	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 arrival time in the past and the timer has gone			<u>1-Inalize Operation constructor with arrival time less than timer</u> <u>2-Inalize 1 Operations with valid input</u>	<u>1-Inalize Operation constructor with arrival time 0 timer</u> <u>2-Inalize 1 Operation with valid input</u> <u>3-call enqueue()</u> <u>4-call consumeTimeUnit()</u> <u>5-call enqueue with the operation arrivalTime less than timer</u>	Throws illegal argument exception	exit	Throw illegal argument exception	Y
19	if the queue in FCFSQ is null and call consumeTimeUnit			<u>1-intalize FCFSQ</u>	1- <u>intalize FCFSQ</u> 2- <u>call consumeTimeUnit()</u>	Return null	exit	Return null	Y
20	Put one operation in queue and consume time unit with			<u>1-intalize FCFSQ</u>	1- <u>intalize FCFSQ</u> 2- <u>inalize Operation with exeTime 1</u>	Return null	exit	Return null	Y

## Testing Type Details: Unit Testing

Tester Name: Group1

### Test Environment Details

Scen #	Scenario Description	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
	exeTime 1				3- <u>call</u> <u>consumeTimeUnit()</u> 4- <u>call</u> <u>consumeTimeUnit()</u> <u>again</u>				
21	Empty iterator in FCFS			<u>1-inalize FCFSQ</u>	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			<u>1-Inalize Operation constructor with arrival time less than timer</u> <u>2-Inalize 1 Operations with valid input</u>	<u>1-Inalize Operation constructor with arrival time 0 timer</u> <u>2-Inalize 1 Operation with valid input</u> 3-call enqueue() 4-call consumeTimeUnit() 5-call enqueue with the operation arrivalTime less than timer	Throws illegal argument exception	exit	Throws illegal argument exception	Y
23	Temp queue is empty			1- <u>inalize SJFSQ</u>	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			<u>1-inalize SJFSQ</u>	1- Inalize SJFSQ 2- Call consumeTime() 3- Call consumeTime()again	Return null	Exit	Return null	Y

## Testing Type Details: Unit Testing

Tester Name: Group1

### Test Environment Details

Scen #	Scenario Description	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
25	We will put all arrivalTime queues more than timer so it will be in Temp queues			1- <u>Inalize SJFSQ</u> 2- <u>Inalize 4 Operations with valid data but arrival time bigger than timer</u>	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- <u>inalize SJFSQ</u> 2- <u>inalize one operation with 1 exeTime</u>	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
27	In PreemptiveSJFQ queue is empty			<u>1-inalize PreemptiveSJFQ</u>	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- <u>Inalize PreemptiveSJFQ</u> 2- <u>Inalize 4 Operations with valid data but arrival time bigger than</u>	1- Inalize PreemptiveSJFQ 2- Inalize 4 operations with valid data but arrival time bigger than timer 3- Enqueue it 4- Call	Return null in first call but second call work normally	exit	Return null in first call but second call work normally	Y

## Testing Type Details: Unit Testing

Tester Name: Group1

### Test Environment Details

Scen #	Scenario Description	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
				<u>timer</u>	consumeTimeUnit( ) two times				
29	One process at queue			1- <u>inalize</u> Preemptive <u>SJFQ</u> 2- <u>inalize one</u> <u>operation</u> <u>with 1</u> <u>exeTime</u>	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
Round Robin									
30	RoundRobin enqueue it when the arrival time in the past and the timer has gone			<u>1-Inalize Operation</u> <u>constructor with</u> <u>arrival time less than</u> <u>timer</u> <u>2-Inalize 1</u> <u>Operations with valid</u> <u>input</u>	<u>1-Inalize Operation</u> <u>constructor with arrival</u> <u>time 0 timer</u> <u>2-Inalize 1 Operation with</u> <u>valid input</u> <u>3-call enqueue()</u> <u>4-call consumeTimeUnit()</u> <u>5-call enqueue with the</u> <u>operation arrivalTime less</u> <u>than timer</u>	Throws illegal argument exception	exit	Throws illegal argument exception	Y

## Testing Type Details: Unit Testing

Tester Name: Group1

### Test Environment Details

Scen #	Scenario Description	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
31	In RoundRobin queue is empty			<u>1-inalize RoundRobin</u>	1- Inalize RoundRobin 2- Call consumeTime() 3- Call consumeTime()again	Return null	exit	Return null	Y
32	We will put all arrivalTime queues more than timer so it will be in Temp queues			1- <u>Inalize RoundRobin</u>  2- <u>Inalize 4 Operations with valid data but arrival time bigger than timer</u>	1- Inalize RoundRobin 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
33	One process at queue			1- <u>inalize SJFSQ</u> 2- <u>inalize one operation with 1 exeTime</u>	1- inalize RoundRobin 2- inalize Operation with valid input but exeTime 1 3- enqueue it 4- ConsumeTimeUnit( ) 5- Call consumeTimeUnit( ) again	Return null	exit	Return null	Y
34	We will put all arrivalTime queues more than timer so it			1- <u>Inalize RoundRobin</u>	1- Inalize RoundRobin 2- Inalize 3	Work normally	Exit	Work normally	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)
	will be in Temp queues and operation in queue will be less round robin but not equal zero			2- <u>Initalize 3 Operations with valid data but arrival time bigger than timer</u> 3- <u>Inaltize operation with exeTime less than quantum but same arrival time in ready queue</u>	operations with valid data but arrival time bigger than timer 3- Enqueue it 4- Inalize Operation with arrival time equal timer with exe Time less than quantum 5- consumeTimeUnit( ) two times				
Same for Preemptive and Priority queues and success									
35	In PreemptivePriority Queue different priority but same arrival time so the highest priority will be on cpu			1- <u>inalize Preemptive Priority queue</u> 2- <u>inalize two operation with same arrival time but different piroity</u>	1- <u>inalize Preemptive Priority queue</u> 2- <u>inalize two operation with same arrival time but different piroity</u> 3- <u>enqueued them</u> 4- <u>check asseration</u>	Highest piriocity on cpu	exit	Highest piriocity on cpu	N

## Testing Type Details: Unit Testing

Tester Name: Group1

### Test Environment Details

Scen #	Scenario Description	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
36	In PreemptivePriority Queue different priority and same arrival time and same priority so shortest job must be first will be on cpu			<u>1-inalize Preemptive Priority queue</u> <u>2- inalize Three operation with same arrival time and same piroity</u>	<u>1-inalize Preemptive Priority queue</u> <u>2- inalize three operation with same arrival time and same piroity</u> <u>3- enqueued them</u> <u>4- check asseration</u>	Shortest job will be on cpu	exit	Highest piroity on cpu	N
37	Same id								
38	In Preemptive Shortestjob first Queue different arrival but same brust after consumeTimeUnit so the earliest arrival Time will be on CPU			<u>1-inalize Preemptive PShortestJob first queue</u> <u>2- inalize three operation with different arrival time but same brust after consume</u>	<u>1-inalize Preemptive SJF queue</u> <u>2- inalize three operation with different arrival time but same brust after consume</u> <u>3- enqueued them</u> <u>4- check asseration</u>	FIRST ARRIVAL STILL ON CPU	exit	FIRST ARRIVAL STILL ON CPU	y
39	In Preemptive Shortestjob first Queue same arrival and same brust but different piroity after consumeTimeUnit so highest piroity must enter			<u>1-inalize Preemptive PShortestJob first queue</u> <u>2- inalize two operation with same arrival time and same brust s</u>	<u>1-inalize Preemptive PShortestJob first queue</u> <u>2- inalize two operation with same arrival time and same brust</u> <u>3- enqueued them</u> <u>4- check asseration</u>	Highest piroity on Cpu	exit	First one with medium piroity	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)
40	Duplicate process and ID			<u>1-inalize 3 operation with same all</u>	<u>1-inalize 3 operation with same all</u>	Illegal argument excpetion	exit	Work normally	N

### Testing Type Details: Unit Testing

Tester Name: Group1

#### Test Environment Details

Scen #	Scenario Description	R e q #	Co nd #	Test Data	Test Conditions/Steps	Expected Results/Comments	Post-Conditions	Actual Results	Pass/Fail (Y/N)
	112								