Test Plans and Actual Test Data

Method	**Test Cases (marks)**	**Actual Test Data**	**Expected Output**
Job.IsValidId	1. Normal valid Id (0.5)	id = 500	True
	2. Normal invalid Id (0.5)	id = -500	False
	3. Boundary valid Id (0.5)	id = 1, 999	True
	4. Boundary invalid Id (0.5)	id = 0, 1000	False
Job.IsValidExecutionTime	1. Normal valid execution time (1.0)	executiontime = 10	True
	2. Boundary valid execution time (0.5)	executiontime = 1	True
	3. Boundary execution time (0.5)	executiontime = 0	False
Job.IsValidPriority	1. Normal priority (0.5)	priority = 5	True
	2. Normal invalid priority (0.5)	priority = 5	False
	3. Boundary valid priority (0.5)	priority = 1, 9	True
	4. Boundary invalid priority (0.5)	priority = 0, 10	False
Job.IsTimeReceived	1. Normal valid time received (1.0)	time = 10	True
	2. Boundary valid time received (0.5)	time = 1	True
	3. Boundary invalid time received (0.5)	time = 0	False
JobCollection.Add	1. Add a job to an empty job	JobCollection1.Add(J1)	True;

	collection (0.5)		new Count = old Count + 1; new Capacity = old Capacity
	2. Add a new job to a job collection that contains one job (0.5)	JobCollection2.add(J1)	True; new Count = old Count + 1; new Capacity = old Capacity
	3. Add a new job to a collection that contains multiple jobs (0.5)	Jobcollection3.add(J1)	True; new Count = old Count + 1; Capacity = old Capacity
	4. Add a duplicate job to a job collection (0.5)	Jobcollection3.add(J2)	False; new Count = old Count; new Capacity = old Capacity
JobCollection.Contains	1. Check if a given job is in an empty job collection (0.4)	JobCollection1.Contains(J1.Id)	False
	2. Check if a given job is in a job collection containing one job and the given job ID matches the job in the job collection (0.4)	JobCollection2.Contains(J2.Id)	True
	3. Check if a given job is in a job collection containing one job and	JobCollection2.Contains(J1.Id)	False

	the given job ID does not match any job in the job collection (0.4)		
	4. Check if a given job is in a job collection containing multiple jobs and the given job ID matches one of the jobs in the job collection (0.4)	JobCollection3.Contains(J2.Id)	True
	5. Check if a given job is in a job collection containing multiple jobs and the given job ID does not match any job in the job collection (0.4)	JobCollection3.Contains(J1.Id)	False
JobCollection.Find	1. Check if a given job is in an empty job collection (0.4)	JobCollection1.Find(J1.Id)	Null
	2. Check if a given job is in a job collection containing one job and the given job ID matches the job in the job collection (0.4)	JobCollection2.Find(J2.Id)	J2 ref
	3. Check if a given job is in a job collection	JobCollection2.Find(J1.Id)	Null

	containing one job and the given job ID does not match any job in the job collection (0.4)		
	4. Check if a given job is in a job collection containing multiple jobs and the given job ID matches one of the jobs in the job collection (0.4)	JobCollection3.Find(J2.Id)	J2 ref
	5. Check if a given job is in a job collection containing multiple jobs and the given job ID does not match any job in the job collection (0.4)	JobCollection3.Find(J1.Id)	Null
JobCollection.Remove	1. Job collection contains no jobs	JobCollection1.Remove(J1.Id)	False
	2. Job collection contains one job and the job is the one to be removed	JobCollection2.Remove(J2.Id)	True
	3. Job collection contains one job and the job is not the one to be removed	JobCollection2.Remove(J1.Id)	False
	4. Job collection is full and the given job is in	JobCollection4.Remove(J1.Id)	True

	the job collection		
	5. Job collection is full and the given job is not in the job collection	JobCollection4.Remove(J9.Id)	False
JobCollection.ToArray	1. Job collection contains no jobs	JobCollection1.ToArray()	Nil
	2. Job collection contains one job	JobCollection2.ToArray()	J2
	3. Job collection contains multiple jobs	JobCollection3.ToArray()	J2, J3, J4, J5, J6
	4. Job collection is full	JobCollection4.ToArray()	J1, J2, J3, J4, J5, J6
Scheduler.FirstComeFirstServed	1. Scheduler contains no job (0.5)	Scheduler1	Nil
	2. Scheduler has only 1 job (0.5)	Scheduler2	J1
	3. Scheduler has multiple jobs that arrived at different time (0.5)	Scheduler3	J3, J4, J2, J1
	4. Scheduler has multiple jobs, some of which arrived at the same (0.5)	Scheduler4	J3, J9, J7, j4, J2, J5, J1, J8, J6, J10
Scheduler.Priority	1. Scheduler contains no job (0.5)	Scheduler1	Nil
	2. Scheduler has only 1 job (0.5)	Scheduler2	J1
	3. Scheduler has multiple jobs that arrived at different priority (0.5)	Scheduler3	J1, J3, J2, J4
	4. Scheduler has multiple jobs, some of which have	Scheduler4	J8, J1, J7, J3, J5, J9, J2, J6, J10, J4

	the same priority (0.5)		
Scheduler.ShortestJobFirst	1. Scheduler contains no job (0.5)	Scheduler1	Nil
	2. Scheduler has only 1 job (0.5)	Scheduler2	J1
	3. Scheduler has multiple jobs that have different execution (0.5)	Scheduler3	J4, J2, J1, J3
	4. Scheduler has multiple jobs, some of which have the same execution time (0.5)	Scheduler4	J5, J4, J2, J7, J10, J1, J8, J6, J3, J9

Notes:

- J1 = (jobId: 523, timeReceived: 90, executionTime: 23, priority: 8)
- J2 = (jobId: 966, timeReceived: 46, executionTime: 15, priority: 2)
- J3 = (jobId: 26, timeReceived: 11, executionTime: 50, priority: 5)
- J4 = (jobId: 553, timeReceived: 35, executionTime: 12, priority: 1)
- J5 = (jobId: 346, timeReceived: 79, executionTime: 6, priority: 5)
- J6= (jobId: 560, timeReceived: 95, executionTime: 47, priority: 2)
- J7=(jobId: 132, timeReceived: 13, executionTime: 18, priority: 8)
- J8=(jobId: 741, timeReceived: 92, executionTime: 37, priority: 9)
- J9=(jobId: 267, timeReceived: 11, executionTime: 50, priority: 5)
- J10= (jobId: 583, timeReceived: 97, executionTime: 22, priority: 2)
- Jobcollection1 contains no job (capacity = 6)
- Jobcollection2 contains only one job, J2 (capacity = 6)
- Jobcollection3 contains jobs J2, J3, J4, J5, J6 (Capacity = 6)
- Jobcollection4 contains jobs J1, J2, J3, J4, J5, J6 (Capacity = 6)
- Scheduler1 contains no job
- Scheduler2 contains J1
- Scheduler3 contains J1-J4
- Scheduler4 contains J1-J10