

Class name: job	Super class: N/A
Subclasses N/A	
Responsibilities job(); - Default constructor job(int, jobType, double, double, double, double, bool); - parameterized Constructor int getJobNumber() const; - Return the Job Number jobType getJobType() const; - Return the Job Type double getTimeRequired() const; - Return the time this job requires double getTimeServiced() const; - Return the time this job has been serviced thus far double getArrivalTime() const; - Return the arrival time of this job double getCompletionTime() const; - Return the completion time of this job bool lastService() const; - Return whether this is the last time this job needs serviced in this timeslice void setJobNumber(int jn); - Set the job number void setJobType(jobType t); - Set the job type void setTimeRequired(double t); - Set the amount time required for this job void setTimeServiced(double t); - Set the amount of time this job has been serviced thus far void incrementTimeServiced() - Increment the time this job has been serviced by .1 void setArrivalTime(double t) - Set the arrival time of this job void setCompletionTime(double t) - Set the Completion time of this job void setLastService(bool l) - Set whether this is the last service for this timeslice	
Collaborations N/A	

Class name: jobQueue	Super class: N/A
Subclasses N/A	
Responsibilities jobQueue(); - Default constructor jobQueue(int newSize); - Parameterized constructor ~jobQueue(); - Deconstructor void enqueue(job j); - Enqueue a job void priorityEnqueue(job j); - Priority Enqueue a job job dequeue(); - Dequeue from the queue bool isFull() const; - Return if the queue is full bool isEmpty() const - Return if the queue is empty int getLength() const; - Return the length of the queue void clear(); - Clear the queue job copyFront() - Returns a copy of the front of the Queue	
Collaborations Contains an object of type 'job'	

Class name: simulation	Super class: N/A
Subclasses N/A	
Responsibilities bool isEmptyinCPUs(jobQueue cpu[]); - Returns whether there is an empty spot in any of the CPU queues job createRandomJob(double currTime, int jobNum); - Create a new job (but don't enqueue it) simulation(); - Constructor void init(); - Initialize the simulation void run(); - Run the simulation	
Collaborations Relies on both th 'job' and 'jobQueue' classes.	

