Test your implementation with a random input that adheres to the criteria above. Submit a writeup of your design, screenshots of the results, and code files in a zip file.

A screen shot of a computer screen

Description automatically generated

**Figure 1.** Demonstration of simulated OS scheduler using randomly generated durations and priorities. Jobs with minimum priority are executed first and may not be interrupted. At each time slice, the user may add another job to the queue.

The implementation of the Scheduler class uses an underlying minimum heap priority queue structure, which is based on a vector-based binary tree. The minimum heap is a template structure which uses a custom Job element as well as a custom comparator, JobComparator. The Job objects have members for a std::string name, an integer duration, and an integer priority, of which the numerical members are validated upon construction where necessary. The Job class also has appropriate getter and setter methods to facilitate interaction with private data members. JobComparator simply accesses the Jobs priority member to determine which job has the highest priority (which has the lowest value [-19:20]) and uses this to restructure the minimum heap as necessary. The Scheduler class itself has methods which allow adding new job objects, printing the entries in the job queue from highest priority to lowest, checking if the scheduler is empty, and running a single time slice. When running a single time slice, the highest priority job has it’s duration decremented by 1 until the job is completed (has a duration of 0), at which point the job is removed from the priority queue entirely. In the demonstration, the user may specify a new job to add to the queue, providing the details necessary to populate the parameterized Job constructor. They may also enter ‘n’ or ‘N’ if they do not wish to add a new job, or simply press enter (submit an empty line). The user may also quit the program at any time they wish. After each time slice is executed, the new duration and/or new job that is loaded for execution is printed to the console. Upon opting to add a new job, the console reports the command sent to the scheduler, either “add job name with length n and priority p” or “no new job this slice” as required by the assignment statement.