Lab 1	10% of Final Grade		
Goals:			
Implement Priority Queue correctly			
**Please assume that the Job with the higher priority is the one with the greater number --> Ex. 1 would be the lowest priority			
**must be turned in as .jar file and take a text file input to run			
POINTS Breakdown:			
No Submission	0	0	
Accepts .txt file:		10	TA can pass in a text file of the same structure as in assignment desctiption
Gives relevant output:		10	Output gives wait time and execution time
Followed good coding standards (proper indentation, comments, spacing):		10	Code is neat and easily readable
Basic class structure is present:		15	Job and job scheduler class present, other classes may be used depending on structure
Runs but no priority queue/heap implemented:		15	Jobs run exactly as they come in not according to priority
Attemp at Priority Queue/heap:		10	Student attemps a priority heap/queue but it doesn't run or order jobs correctly
Implements Priority Queue/heap:		10	Priority heap/queue is implemented but may not run with full correctness
Runs with minor error:		10	One or two jobs out of order. Minor code errors other places
Everything Correct		10	Good Job!
Bonus - multiple processors		5	
Final		100	105 Points Possible

Way grading will be done:
Grading in Lab (will give maximum chance at points because you can walk TAs through your code and logic):
TA will give students specific input file to run and student(s) give a quick run down of code structure. If output looks good student is checked off. If errors are present TA will record grade based on those errors, and Student(s) will be able to fix code and show to TA later in lab or can submit error free code to D2L and TA will run code to see if higher grade is warranted (See below for how D2L submission will be geraded).
Grading from D2L:
TA will pass an input file to .jar file and look at output. If output is correct TA will then go through code to check organization and make sure it was not copied and full credit is given. If output is not correct see rubric above for how points will be awarded.
Things to keep in mind:
It is hard to grade code line by line. If you want maximum chance at points make it easy for TAs to see what is going on in code. Provide comments descriptive variables, etc. The more you give us to work with (either explaining your code to us in class or with comments in code) the more points we can give cause we won't be guessing at your understanding.