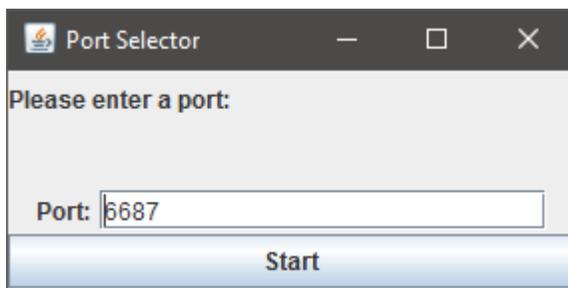


Programming Exam #2  
CSCI 201L Spring 2016  
10% of course grade

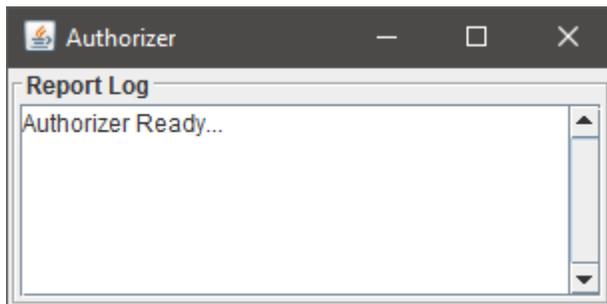
## Part 1 – Authorizer

Before workers can begin working, they must be authorized through the Factory Authorizer program. You will be given this program in the form of a .jar, along with some guidelines and hints on how to communicate with it. You will also be given a Decoder.jar, which can be used to decode the messages. Add this jar file to your build path so you can call the appropriate method.

Start the authorizer program by double clicking the FactoryAuthorizer.jar file, then run it on any port and click start.



Once started, you will see the following:



The authorizer uses a `BufferedReader` and a `BufferedWriter` to read and write strings.

The authorizer waits for connections from workers. The program can communicate with an unlimited amount of workers at a time. Once it receives the worker's name, it will generate and send 20 random strings to the worker. The strings must be decoded by the worker and sent back to the authorizer within 10 seconds.

The time to decode a string is about 2.5 seconds. Strings will need to be decoded separately, and then appended to one another in the order they were received.

Here are some sample decodings. Note: these are not actual results.

ABCDE	11011
FGHIJ	00100
KLMNO	10101
PQRST	01010
UVWXY	11011

So for example, if the worker receives:

FGHIJ

UVWXY

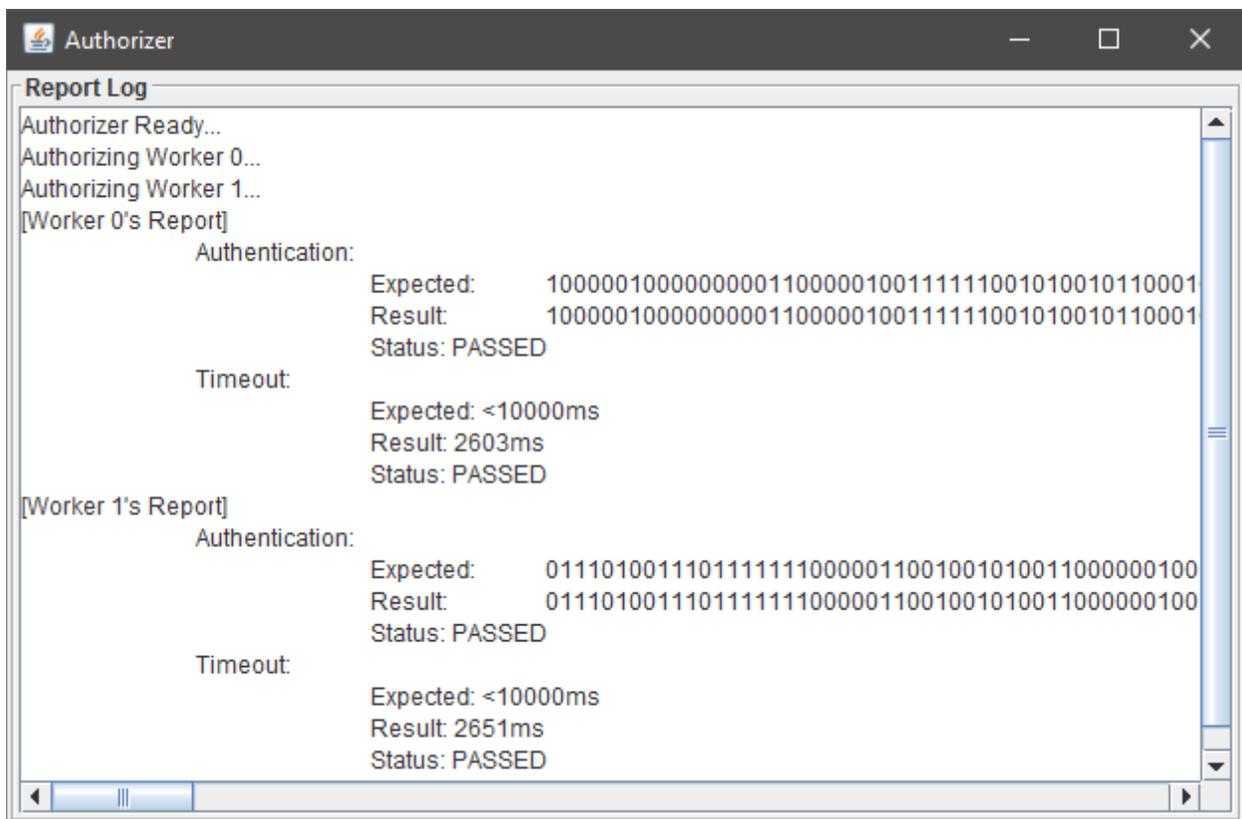
ABCDE

PQRST

The worker will need to send back 00100110111101101010

Keep in mind that *String Decoder.resolve(String)* does all of the work. There is no decoding puzzle for you to solve. You just need to append the results.

This is an example of a successful authorization of two workers:



For full credit, you must get 'PASSED' for both 'Authentication' and 'Timeout' for all workers.

Sending a bad authentication result under 10 seconds will award partial credit.

Sending a correct authentication result over 10 seconds will award partial credit.

## Part 2 – Toolboxes

Add Toolboxes to the factory. The coordinate and amount will be dictated by the server.

```
9 -- these are all of the resources
10 Resource|Motherboard|500|9|8
11 Resource|Processor|800|3|2
12 Resource|Hard Drive|900|1|6
13 Resource|Memory|800|3|11
14 Resource|Box|500|5|4
15 Resource|Toolbox|3|12|8
16
```

For testing purposes, set the number of toolboxes to be less than the number of workers.

In order to work on a task in the factory, a worker must pick up a toolbox. After the worker gets assigned a task at the task board, the worker will pick up a toolbox if one is available. If there are no toolboxes available, the worker will have to wait until a toolbox is returned. Toolboxes are returned right before a worker completes their task at the task board.

Worker about to take the last toolbox:



Workers waiting for more toolboxes to arrive:



Worker put back a toolbox:



There are ways to complete this section without proper programming practices. If you complete this section without them, you will not receive full credit. More often than not, you won't come across a race condition, but graders will mark down your code if race conditions are possible.

% of final grade	Criteria
1.0%	Workers connect to authorizer with their name.
2.5%	Expected authorization results are achieved.
2.5%	Expected timeout results are achieved.
1.0%	Toolbox coordinates and amounts are dictated by the server.
1.0%	Workers behave correctly with toolboxes.
2.0%	No potential race conditions with toolboxes.