18-842: Distributed Systems **Spring 2013**
Lab 1: Clocks
Grading Scheme

Demonstrations: Students will need to demonstrate an application program that exercises their ClockService class in both Logical and Vector modes. Note that this application doesn't particularly need to be a distributed app. In particular, they should be able to show the following test cases:

1. Logical clocks are generated properly, always incrementing for each time stamp generated. When a timestamp is received from elsewhere and given to the ClockService, the local timestamp must be advanced to max {local timestamp, remote timestamp}. Timestamps can be compared properly -- test for cases of a ↛ b, b ↛ a and a ll b.
2. Vector clocks are generated properly, always incrementing for each time stamp generated. When a timestamp is received from elsewhere and given to the ClockService, the local timestamp must be advanced to max {local timestamp, remote timestamp} for each element of the vector. Timestamps can be compared properly -- test for cases of a ↛ b, b ↛ a and a ll b.
3. Messages sent and received with MessagePasser are properly integrated with the ClockService (i.e. new timestamp on sent messages, timestamps potentially advanced with received messages)
4. Show a logging facility. This should be a different application, using MessagePasser, to receive messages and track their order in a ↛ fashion. Students should be comparing the timestamp objects, not actually inventing their own way of doing the comparison.
5. Discuss performance in the presence of errors. Students should be able to talk about failure modes of the network (i.e. drop/delay/duplicate) as well as the processes.

Going Forward: Make certain that all errors in the ClockService and associated classes need to be fixed quickly, as future labs will depend on these clocks.