18-842: Distributed Systems **Spring 2013**
Lab 0: Communications Infrastructure
Grading Scheme

Demonstrations: Students will need to demonstrate an application program that exercises their MessagePasser class. At a **minimum**, they should be able to show the following test cases:

1. Set up 4 processes on at least two distinct machines and show that at least one message can be passed to and from each process (i.e. P1 sends a message to P2, P2 sends a message to P3, P3 sends a message to P4 and P4 sends a message to P1)
2. Show that a message with a particular id is dropped, delayed and duplicated (one at a time) based on entries in the configuration file. For instance, P2 sends a message with id="6" to P4. The config file contains "SendRules: - Action : duplicate - ID: 6" and you can show that P4 gets two copies of the message. Show that the filtering works at both the sender and the receiver. Show that it works *only* at the sender or receiver.
3. Show that a message with a particular kind is dropped, delayed and duplicated (one at a time) based on entries in the configuration file. For instance, P2 sends a message with kind="HTTP_REPLY" and id="7" to P4, followed by second message with kind="IRRELEVANT" and id="8". The config file contains "SendRules: - Action : Delay - Kind : HTTP_REPLY" and you can show that P4 receives msg id=8 before it receives msg id=7.
4. Make sure that reception can be interleaved from multiple sources. For instance, P1 sends a message whose kind will get it delayed at the receiver, P2. If the next message is sent from P3 to P2, does P2 get it before the message from P1?
5. Make sure that the Nth and EveryNth rules works properly. This is a bit complex, as rule processing isn't explicitly visible to the user. Using the example from the lab handout might be good. Send a series of messages from Charlie to Alice and see if the third message gets duplicated. Change the config file from Nth to EveryNth and see if every third message gets duplicated.

Going Forward: Make certain that all students understand that errors in the MessagePasser class need to be fixed quickly, as future labs will expect to have a perfectly working infrastructure.