Team members:

1. Kiran Rohankar (UFID: 9199 6253)
2. Tejas Bondre (UFID: 4897 1562)

What is working:

Log files are automatically generated in real-time when an application runs. Each actor generates a separate log file which has a trace of all the message exchanges pertaining to that actor. Message sender, receiver, timestamp and actual message are recorded.

For the Bonus part, the actors inherit a trait that makes the logs generate automatically everytime a message is sent, without the need to litter the actual actor itself with the logging code.

How you integrated logging with the actors:

While sending a message, each actor calls “writeMessage” method in the class “MyLogger”, which generates log files for each actor and appends it with the information of each message pertaining to that actor.

For the Bonus part, each actor uses “send” method of its trait MyLogger instead of ! operator. This will cause the message to be logged into the log file of that actor and then sent to intended receiver actor.

What logfiles are produced by the `example.scala` and how to interpret them:

To test this logging scheme, we have built a network of ten actors which send message to any randomly selected actor. This is an implementation of Gossip protocol in full topology with ten actors. For each actor, the logfile of its messages is created in the same project directory. The logs contain the following information: id of the sender actor, id of the receipient actor, timestamp and the message.