# Grid Computing M (GC-M)
# RMI Assessed Exercise: 2006–07

*Issued: Tuesday 23rd January 2007*
*Submission deadline: 11am Tuesday 30th January 2007*

This is the second of the assessed warm-up exercises for the module, and it builds on the unassessed activity issued at the second lecture of the module, on Java RMI. The exercise will be marked as either 0 or 1. Your overall coursework mark for this module is the product (not the sum) of the marks from the warm-up assessed exercises and the main assessed exercise, so it is important that you get a mark of 1 for this exercise. Any reasonable attempt at this exercise will gain 1 mark, non-attempts score 0, truly disastrous attempts may score 0, or trigger additional coursework to again be marked from 0 or 1 in lieu of this exercise.

The exercise is designed to take between 30 minutes (for experienced RMI programmers) and 2 hours (for those new to concurrent and distributed programming), and assumes some background in Java programming.

**Warning:**

This is a solo exercise and must not be discussed with your classmates or anyone else. However, you can discuss the non-assessed code distributed previously, to ensure that you understand the ideas behind distributed programming with Java RMI.

## Problem Statement:

You are to use Java and Java RMI to produce a collection of interfaces and classes which combine to produce a simple distributed system, with three distinct sorts of processes in it. The system should behave as follows:

The user can launch a Logger program, which registers a remotely invocable object in the local rmiregistry. That object supports a `log` method, which takes a String as argument and outputs the string to standard output. In addition, the Logger should retain a record of the last 20 messages output, in the form of a circular buffer of entries which contain the message output and the date and time at which this was done.

The user can then run one or more copies of a Paper-Scissors-Stone program (PSSP), on different machines. This program takes the name of the machine on which the Logger is running as a command line parameter. The PSSP acquires a reference to the Logger object and uses it for all of its output. The PSSP program runs 50 iterations of the childrens game[1] of the same name, using two random numbers to determine which of the three options each of two players has selected and determining whether the first or the second player is the winner, or whether the round is drawn. The PSSP reports, via the Logger, the overall outcome, indicating the number of rounds won by each player and the number of drawn rounds.

Finally, you must also provide a LoggerMonitor program. Again, this takes the name of the machine on which the Logger is running as a command line parameter. When run, this should contact the Logger, acquire a copy of the retained information (i.e. the last 20 timestamped messages), and output that information to standard output.

You are advised to remotely access the Linux boxes in BO709, using ssh, as a pool of machines to allow you to run your processes on different machines. Ensure that your programs behave appropriately in the event of any difficulties, such as failure of the Logger, and ensure that appropriate concurrency control is used.

## What to Submit:

Your submission should consist of hardcopy printouts of all of your .java files (use a2ps to create them), together with a single hand-written piece of paper containing: your matriculation number (but not your name), a brief status report indicating whether or not your solution works, and a short statement (max 2 paragraphs) explaining how you know what the status of your solution is. All of the sheets of paper should be stapled together, along with a “pink plagiarism sheet” (available from the Teaching Office) as the topmost document. Submit the bundle via the appropriately labelled locked box outside the teaching office on the first floor of Lilybank Gardens. Do not use plastic covers or other binders, just staple the papers together. If you attach an A4 envelope with your name on it to the back of the bundle, this may prove useful in the event of any difficulties in returning your work.

[1] In this two player game, each player selects one of the three options (Paper, Scissors, Stone) and they compare choices. Paper is beaten by Scissors which is beaten by Stone which is beaten by Paper; if both players choose the same the game is a draw.