# UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

## SOFE 4790U: Distributed Systems (Fall 2018)
## Instructor: Dr. Q. Mahmoud

### Individual Programming Assignment #2

The objective of this individual programming assignment is to get a flavour of the effort involved in designing and developing distributed applications using Java RMI. You will practice designing and developing an innovative distributed application of your choice using Java RMI.

**The Task**
Design and develop a novel, easy to use, and useful distributed application of your choice. It must accomplish something useful and has some novel features. Your Java RMI remote interface must provide 5 unique methods (services) to clients.

**Guidelines**
- Your application should continue to handle clients' requests until it is manually terminated, or you have a UI for run/shutdown.
- If during the demo, your application hangs in the middle of a connection, you will have a second chance to demo again (must be within 5 minutes). If this behaviour persists, it means your application is not completely functional – could be due to issues with multi-threading, and you will lose 75% of the whole assignment mark. Your application must be fully functional in order to get reasonable marks on the other assessment items (see grading rubrics on next page).

**Important Notes**
- You need to demo your working client-server application (7 minutes) to the Teaching Assistant (TA). Email him at Ahmed.Badr@uoit.ca to book an appointment.
- Deadline: Assignment#1 must be demoed and submitted **by 12:00pm (noon) on Fri, October 26**. No extensions no matter what is the reason, so plan accordingly.
- Your solution must be designed and developed by yourself (your own work).
- While students are encouraged to discuss the assignment and general ideas for solutions, each student must design and develop his/her own assignment and code. No code sharing allowed, and no students will have the same application.
- The assignment will be assessed based on the grading rubrics provided on page 2 of this document.

**Submission Guidelines**
Submit the following **by 12:00pm (noon) on Friday, Oct 26**.
1) Source code of your solution – must be submitted on Github – go to the following link for Assignment2: https://classroom.github.com/a/tQmyIkCS
2) The report must be submitted through Blackboard (look under Content -> Assignments –> Assignment #1 Report Submission). It must be in PDF or Word and must be one full-page (approx. 500 words) detailing your application, novel feature(s), challenges and solutions. You may include one clear diagram but no screenshots of the application.

**Grading Rubrics**

| Item (%) | Excellent (full mark) | Good (75%) | Satisfactory (50%) | Unsatisfactory (zero) |
|---|---|---|---|---|
| **Functionality (20)** | Fully functional with no errors or warning. | Functional but nothing special about it and sometimes no response. | Basic functionality beyond code covered in class. | Doesn't compile or run. |
| **Usefulness and usability (20)** | Useful and intuitive to use | Works but nothing special | Requires a manual to use. | Non-existent. |
| **Novel features (20)** | Creative and offers novel functionalities. | Doesn't implement sufficient unique functionalities. | Nothing special. | Non-existent. |
| **Report (20)** | Clearly documented and well organized with novel feature, challenges and solutions. | Readable but not well organized or missing parts. | Documentation is brief. | No documentation. |
| **Source code (20)** | Follows coding standards (name, date, title, meaningful variable names, whitespaces, etc.) and code is fully documented. | Readable source code. Doesn't follow coding standards. | Spaghetti code. | No source code provided or the link to the source code is not accessible. |