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

Individual Programming Assignment #1

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

The Task
Design and develop a novel, usable, and useful client/server application of your choice. It must accomplish something useful and has some novel features. It must provide at least 2 significant functionalities or services to clients, and must have 2 novel features.

Guidelines
- Your application must consist of a multi-threaded server and a client, and it cannot be an HTTP proxy server.
- You must use the Java programming language.
- You must use sockets; applications that make use of high-level APIs (such as URL, URConnection, etc) will not be accepted.
- 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 Friday, October 5**. 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 5**.
1) Source code of your solution – must be submitted on Github as per the following instructions:

     a.   Create a Github account (https://github.com/join)
     b.   Register for Github Devpack in order to get private repositories
         (https://education.github.com/pack)
     c.   Go to the following link for Assignment1: https://classroom.github.com/a/MJV-PsTr
     d.   For additional instructions on how to use Github and submit your assignment, go to the
         following link: https://github.com/UOITEngineering/student-classroom-assignments

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.	One novel feature.	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.