Project 2
Due: Saturday, May 14th

This project must be done individually. There will be no exceptions. The purpose of this project is to develop a client/server version of Project 1, a simulation of Remote Procedure Call communication (RPC).

The server will be multithreaded. The main server's thread takes care of the connection requests (establish rendezvous). The spawned server-threads will carry out the extended rendezvous with the clients. Most of the implemented code will be on the server site.

On the client side, you will create the different types of client threads (passengers, cars, and controllers) that will execute concurrently. These clients will ask the main server's thread to establish a connection.

When the connection is accepted by the server, the main server will create another "client helper" thread that will carry out the two-way communication with the client thread. The client will ask the corresponding "client helper" thread to execute sequentially the methods that were implemented in Project 1 as part of the run method. Before each method can be executed, it will send the server a message containing its name and the method name/number to be executed. This can be implemented in different ways. One way (but not the only way) would be to use a switch-case structure. This is similar to the process of creating stubs in the client and server sites.

Note that each run method should contain at least 3 methods. Some of you did not have a good grasp of Object-Oriented Programming (OOP) design knowledge and had only one significant method in the run method. You will have to break down each thread's run method into at least 3 methods.

If your Project 1 hangs and you cannot fix it, comment out the code that creates the problem.

I am attaching a partial example of a previous project based on a different story. It is not the only way it can be done. Please do not ask me to explain the emailed code to you. If it does not make sense, write your own way of doing it (as long as it follows the project description).

Test your solutions by deploying the client site code in one bird and the server site code in a different bird.

Good luck !!!!!!!!!!!!!!