**TCP Chat Room**

**Project out of 85 points**

**~~(20 points):~~** ~~As a user, I want to be able to chat with another person over the local network, so that I can keep in communication with friends and family.~~

**~~(10 points):~~** ~~As a developer, I want to implement the observer design pattern, so that I can send out a notification to all users that a new person has joined the chat room.~~

**~~(10 points):~~** ~~As a developer, I want to implement dependency injection for logging, so that I can log all messages, log when someone joins the chat, and log when someone leaves the chat.~~

**~~(10 points):~~** ~~As a developer, I want to use a dictionary, so that I can store the users in my chat program.~~

**~~(10 points):~~** ~~As a developer, I want to use a queue, so that I can store and process incoming messages.~~

**~~(10 points):~~** ~~As a developer, I want to use C# best practices, SOLID design principles, and good naming conventions on the project.~~

**~~(5 points):~~** ~~As a developer, I want to make good, consistent commits.~~

**(10 points (5 points each))**: As a developer, I want pinpoint at least two places where I used one of the SOLID design principles and discuss my reasoning, so that I can properly understand good code design.

**Chatroom Class** is a good area of solid design within our chatroom app. It has a single responsibility of being a chatroom and all the properties and methods that a chatroom needs, but no more. One of it’s member variables uses an interface as the type (Recipients), allowing us to add either a client or a server and deliver messages to both. The class honors the interface segregation of solid as it does not inherit unused variables or methods and it does not have unused methods required by an interface.

**Message Class** is of solid design. It has a single responsibility – be a message. It honors dependency inversion as it is dependent upon the client being injected at time of creation.

**(Bonus 5 points):** As a user, I want the ability to send and receive direct messages, so that I can choose a specific user to talk to.

**~~(Bonus 5 points):~~** ~~As a developer, I want the ability to create private chat rooms, so that users can join a channel for themselves.~~

**(Bonus 5 points):** As a developer, I want to implement a Graphical User Interface (GUI), so that my users don’t have to do everything in the console.

**HINT:** Use TCPclient instead of raw socket class. This will make things a little bit easier.

**HINT:** There should be two projects in the same solution: one client-side (Client) and one server-side (Server)