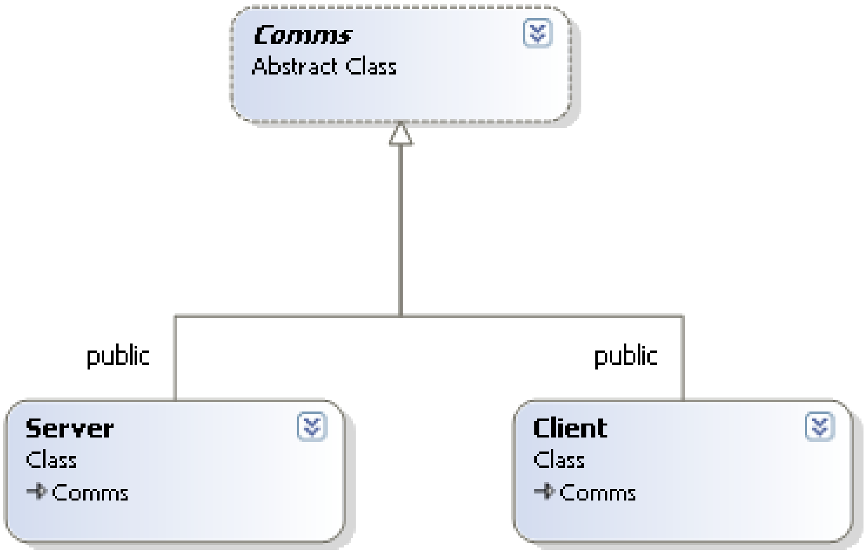
Task 2

The network code developed in the practical sessions took a non-object-oriented approach, with most of the statements placed within the main function. This task requires that you refactor this code into three classes (see class diagram below). The comms class containing the code that is common to both the client and server.
Note: The diagram doesn't show the data members or member functions which you must implement.



The classes should contain the functionality required to establishing a **TCP** WinSock2 connection between two applications running within two processes on a single machines, one being a client and the other a server.
They should also implement the functionality that will allow messages (array of char) to be exchanged between the Client and the Server.

Using these classes develop a chat application that when run, requests the client enter a message which is then echoed on the server interface. A message can then be typed into the server interface and this is then echoed on the client interface. This process should repeat indefinitely until the client types QUIT, at which point both the client and server programs should terminate.

The classes and application should include exception handling code, that replaces much of the original error handling statements found within the practical exercises.

Pre-processor directives should be embedded within your project to enable / disable compilation of client and server code. This should allow you to develop a single project that contains both the Client and Server code.

Your submission should include two projects, one configured to be a server and another client.