# Assignment 1 - Synchronous Chat Program

Assignment Weight: 15% of overall course marks

Submission Date:

**Nov. 5th @ 11:59 pm**. But it is still opened to November 18th. To get benefit from this flexibility, send me email by Oct 25th to specify the submission date that you are going to be able to submit this assignment without asking for any extension within the period from Nov 5th to Nov 18th.

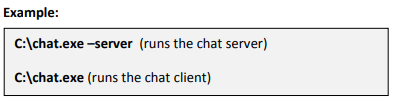
Our next assignments’ due dates are on their regular dates as planned; this means that there might be an overlapping between the due dates of assignment 1 & 2.

**Instructions:**

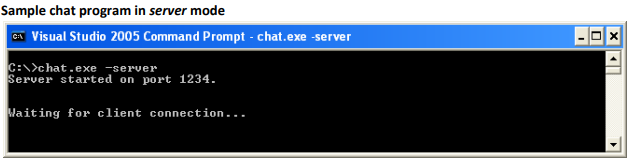
Watch the posted walkthrough video on Teams.

For this assignment, you are tasked to create a simple, console‐based, chat program that will allow the user to communicate, over TCP/IP, to another instance of the chat program.

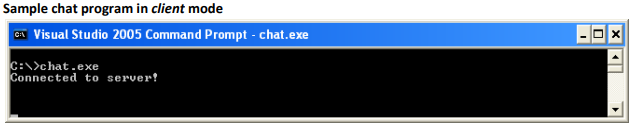
The program will consist of two parts, the server and the client. When the application is run with the "‐server" parameter, the program will run in server mode. When the application is run **without** any parameters, the program will run in client mode.



When the chat program is run in server mode, the application will wait for connection from an instance of the application running in client mode.



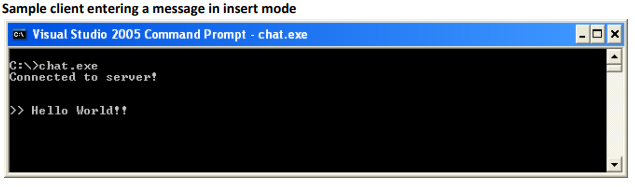
When the chat program is run in client mode, the application will try and connect to the server instance.



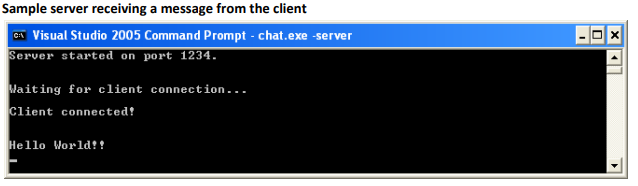
**Note:** *For the purposes of this assignment, you can assume both the server and client are running on the same machine and that only one client can connect to the server at a time.*

Once the client has connected, you can begin sending messages from either the server instance or the client instance.

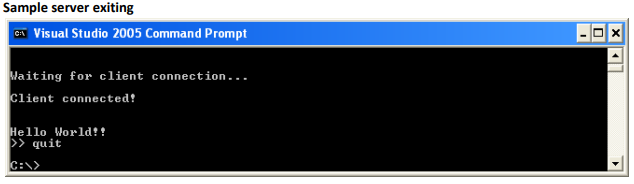
To begin sending messages, you will press the **I** key on the keyboard which will put the chat program into "Input Mode" and display "**>>**" as a prompt. Once in "Input Mode", the user can type a message and it will be sent when the user finishes typing and presses the enter key.



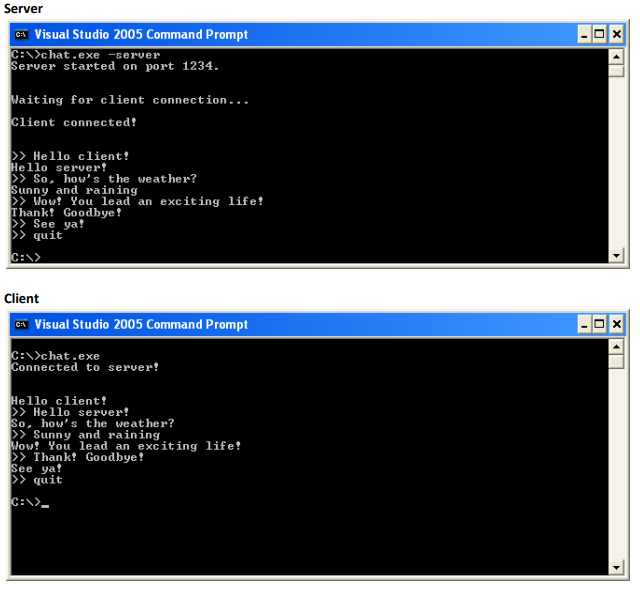
When either server or client is **not** in "Insert mode", the program will be constantly checking for input from the other program. (e.g. the client checks for message from the server and vice versa)



The program will exit when the word "quit" is entered as a message.



**Sample chat session**



Assignment Rubric: Will be posted in another word document file