

Assignment – Synchronous Chat Program

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.

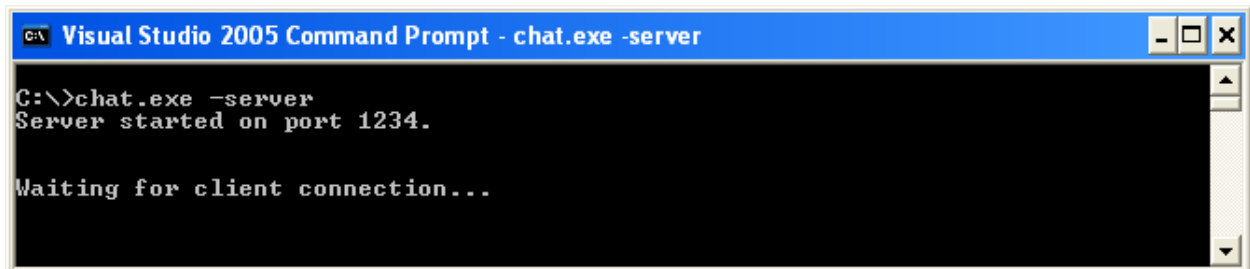
Example:

```
C:\>chat.exe -server (runs the chat server)
```

```
C:\>chat.exe (runs the chat client)
```

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.

Sample chat program in *server* mode

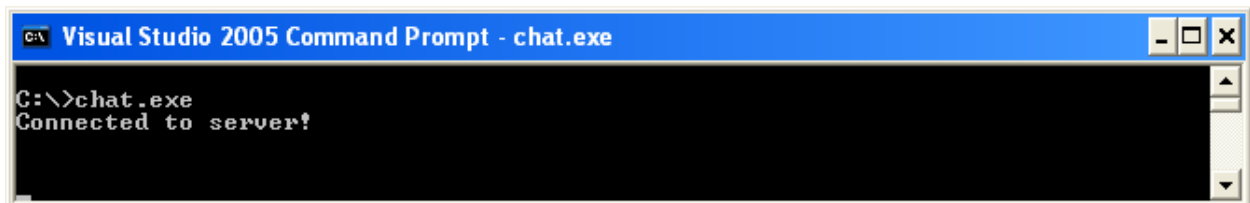


```
C:\>chat.exe -server
Server started on port 1234.

Waiting for client connection...
```

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

Sample chat program in *client* mode



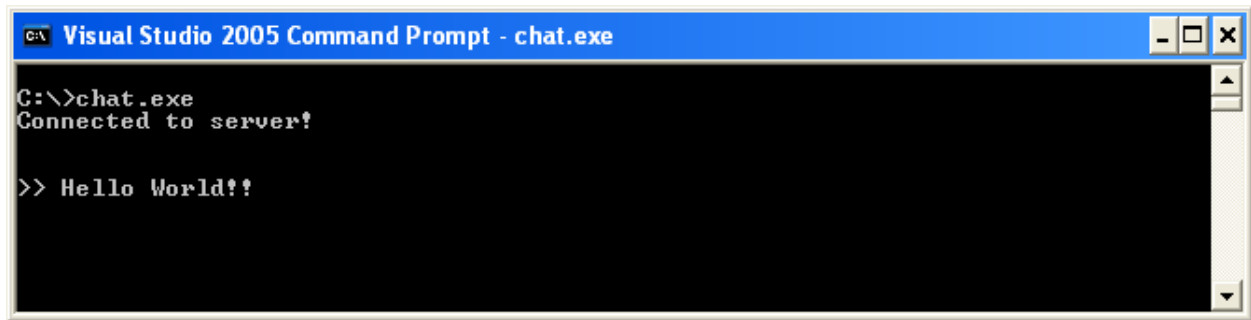
```
C:\>chat.exe
Connected to server!
```

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.

Sample client entering a message in insert mode



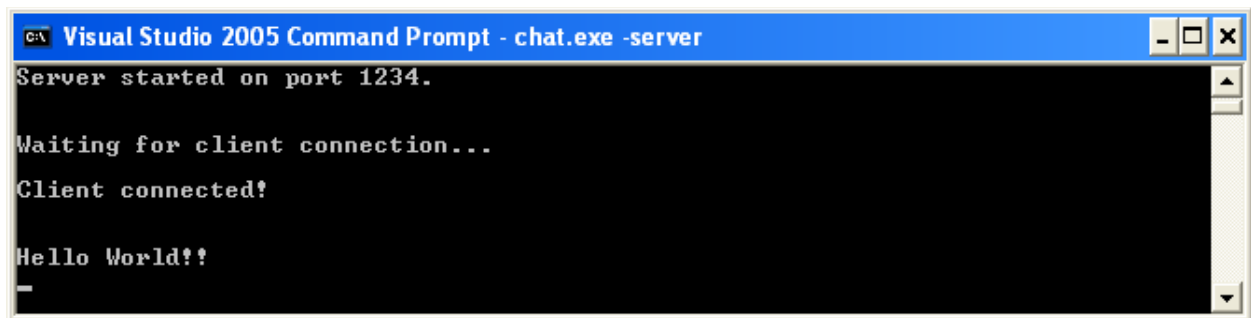
```
C:\> Visual Studio 2005 Command Prompt - chat.exe

G:\>chat.exe
Connected to server!

>> Hello World!!
```

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)

Sample server receiving a message from the client



```
C:\> Visual Studio 2005 Command Prompt - chat.exe -server

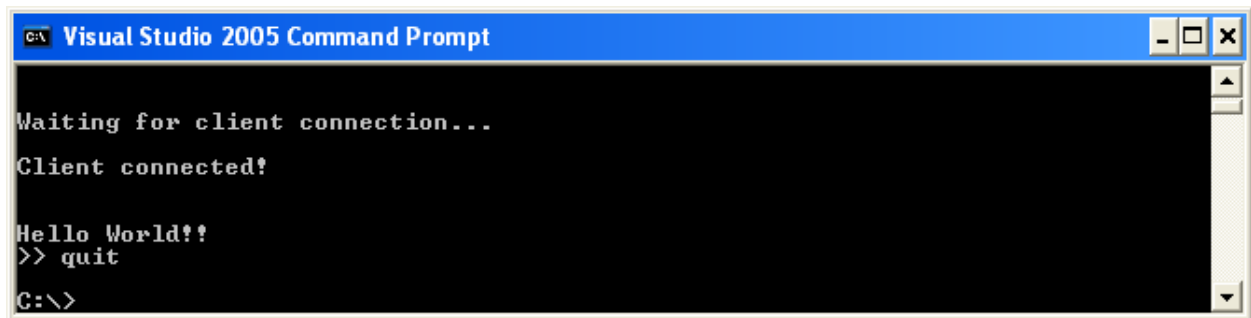
Server started on port 1234.

Waiting for client connection...
Client connected!

Hello World!!
```

The program will exit when the word “**quit**” is entered as a message.

Sample server exiting



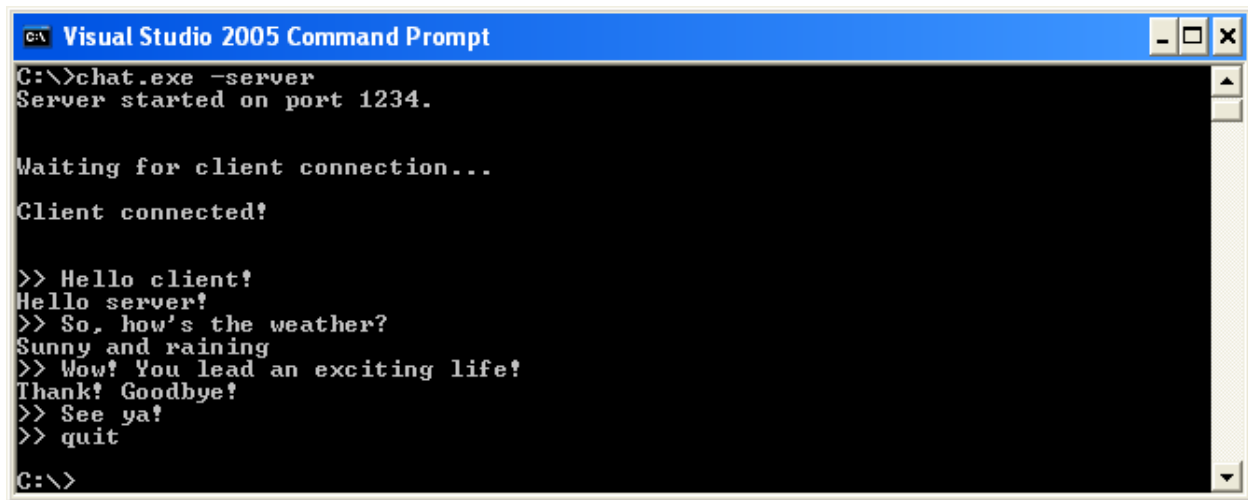
```
C:\> Visual Studio 2005 Command Prompt

Waiting for client connection...
Client connected!

Hello World!!
>> quit
C:\>
```

Sample chat session

Server

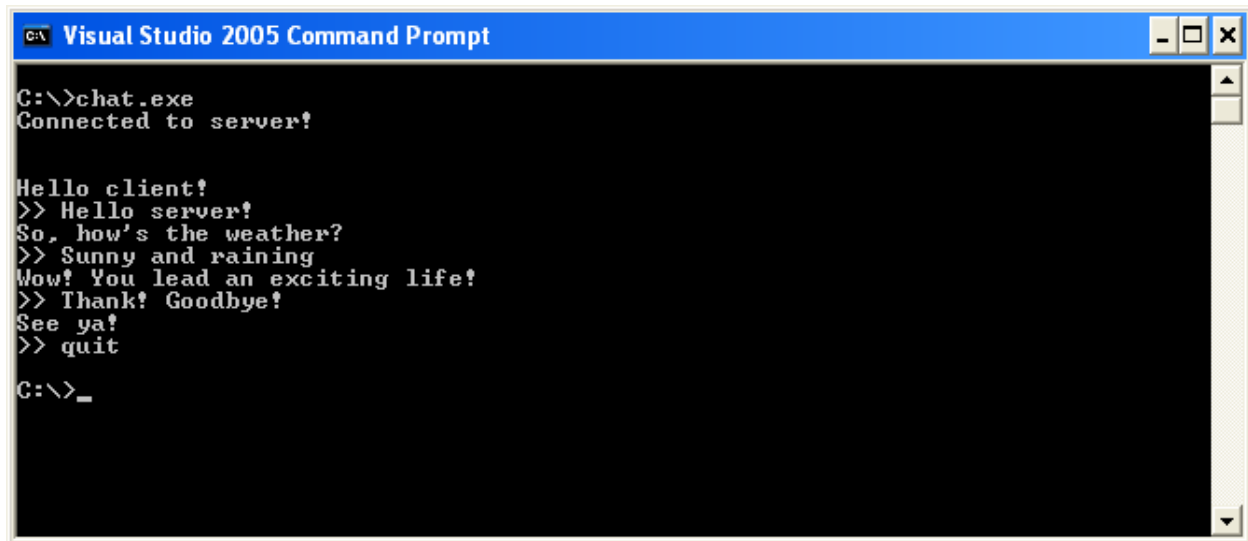


```
C:\>chat.exe -server
Server started on port 1234.

Waiting for client connection...
Client connected!

>> Hello client!
Hello server!
>> So, how's the weather?
Sunny and raining
>> Wow! You lead an exciting life!
Thank! Goodbye!
>> See ya!
>> quit
C:\>
```

Client



```
C:\>chat.exe
Connected to server!

Hello client!
>> Hello server!
So, how's the weather?
>> Sunny and raining
Wow! You lead an exciting life!
>> Thank! Goodbye!
See ya!
>> quit
C:\>_
```