**Lab#2**

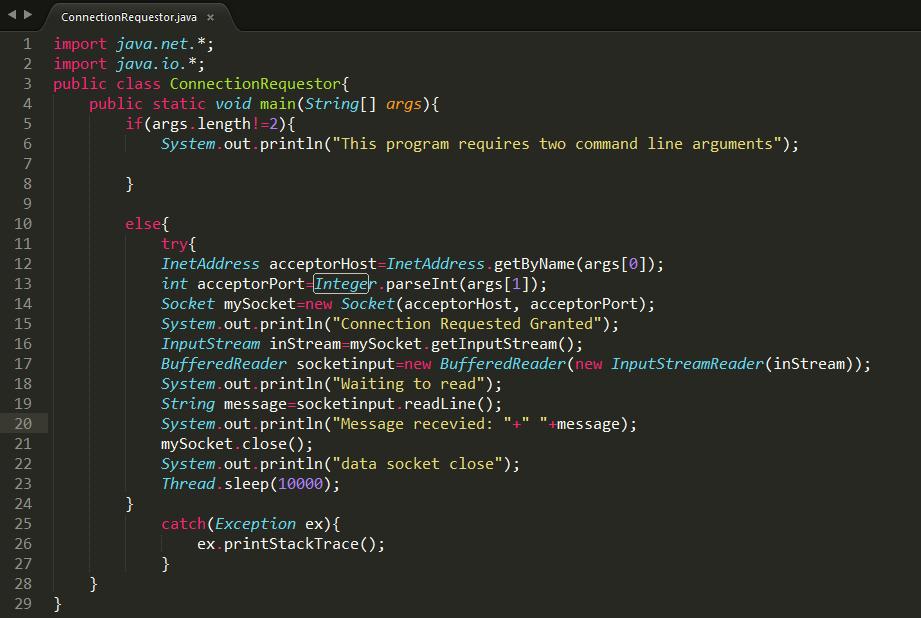
Objective:To become familiar with stream sockets

Class Task#01 And 02:

Compile and run the above code. Start the acceptor first and then the requestor with appropriate command line arguments. Describe and explain the output.

Now run the code again, but reverse the order of program’s execution. Start the requestor first and then the acceptor. Describe and explain the outcome.

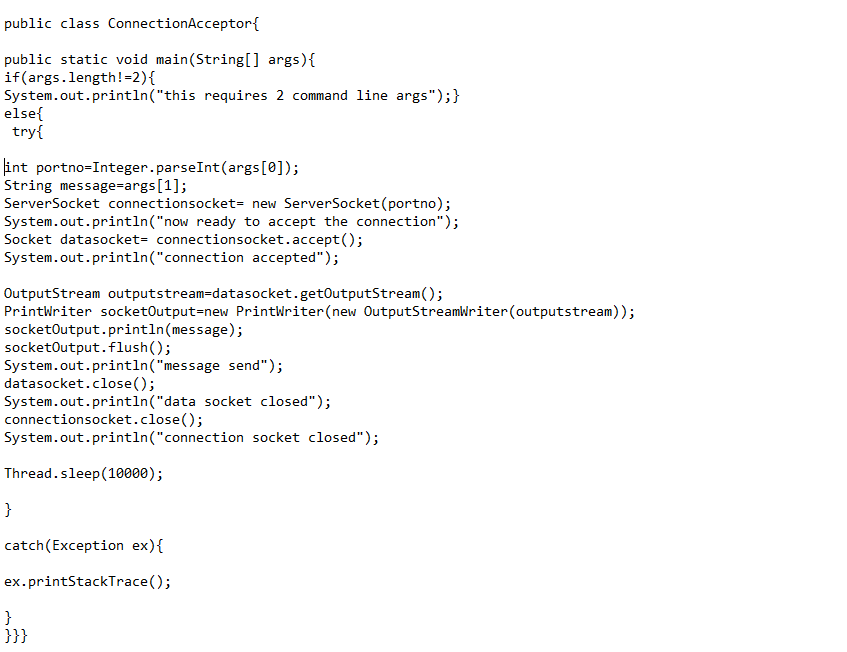
Client

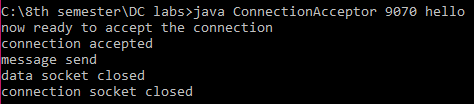


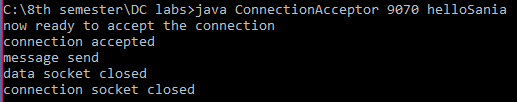




Server



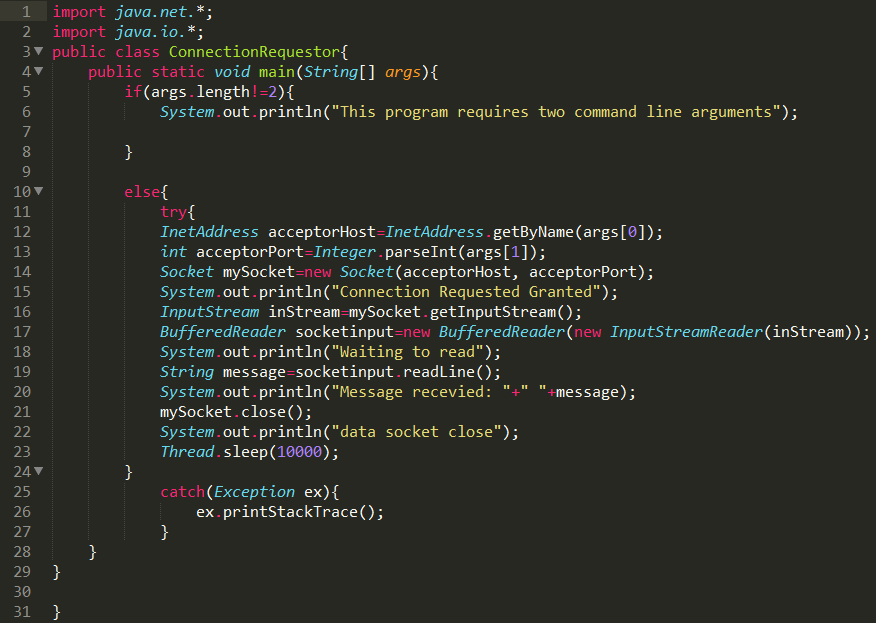




Class Task#03:

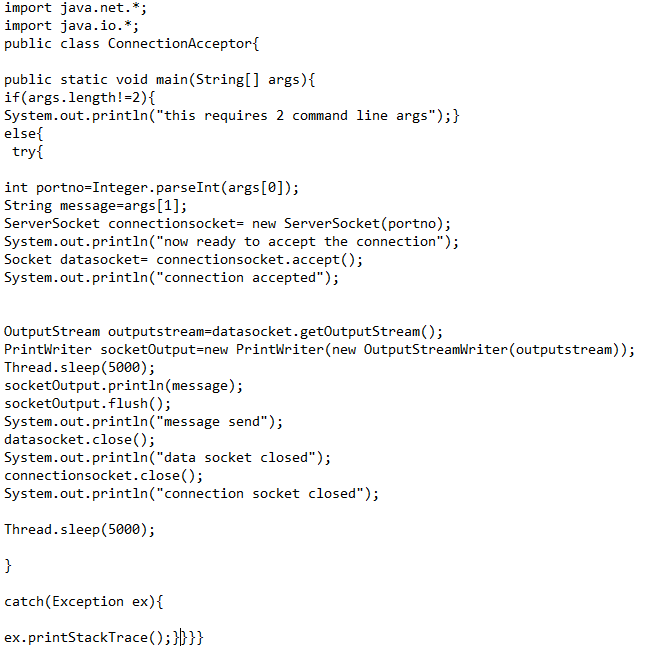
Add a time delay of 5 seconds in the ConnectionAcceptor process just before the message is written to the socket, then run the program. This will show you the blocking at the receiver. Show a trace of the output of the processes.

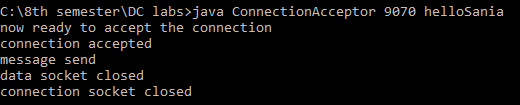
Client





Server



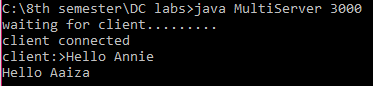


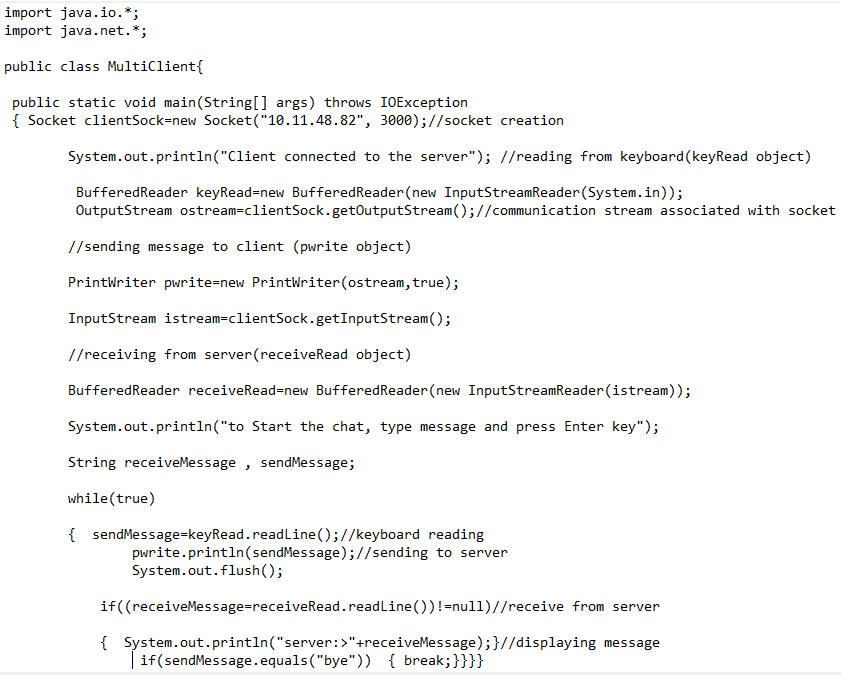
Bounus Task#01:

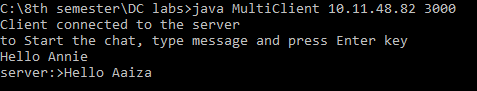
Modify the sample code to include two way communication between the client and the server.

Client And Server





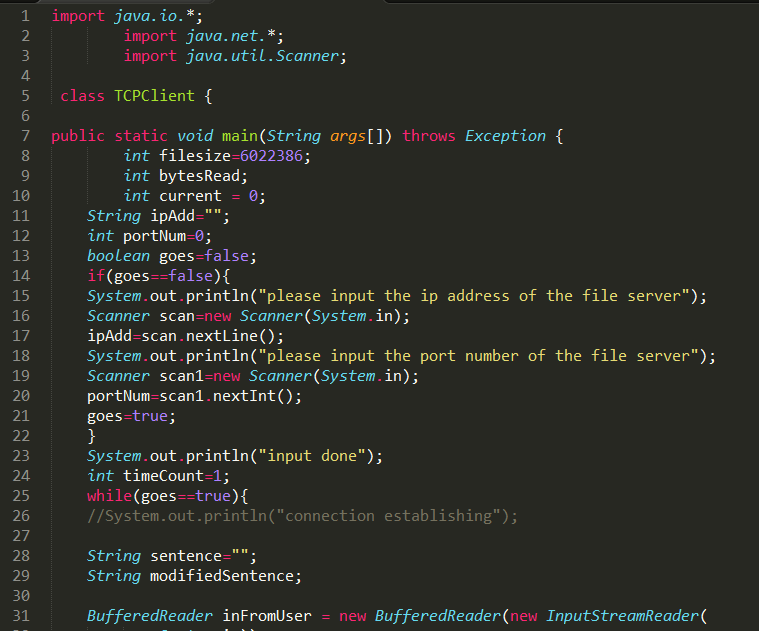




Bounus Task#02:

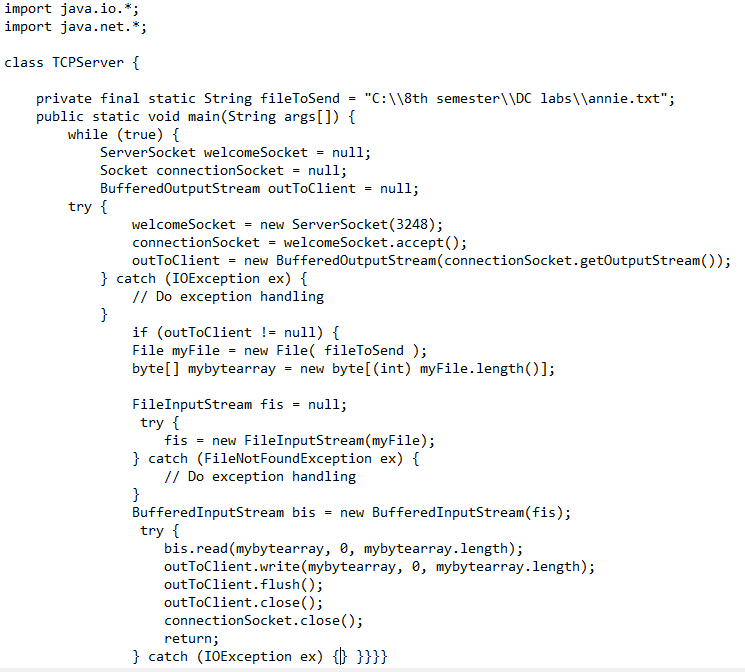
Modify the sample code to send complete files between the client to the server.

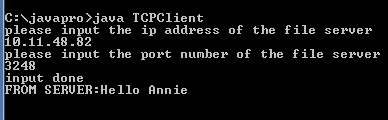
Client And Server











Bounus Task#03:

Explore the non-blocking java socket API in the **nio** package and implement a sample program.

Client And Server



