Assignment, simple chat

You can use the logic of SimpleClient and SimpleServer as starting points to implement a simple chat. We want two applications that can alternatingly send messages between each other. A message is a single string.

1. As things are, SimpleClient only reads, and SimpleServer only talks through the network. In order to do both, each application needs a BufferedReader and also a PrintWriter. To allow input they also need a Scanner each. You will create two programs, SimpleChatServer and SimpleChatClient.

2. The communication has to be strictly ordered: when the client talks the server has to listen and vice versa. Let's implement this with the server starting to talk while the client listens. After one such exchange they reverse roles, the server listens and the client talks. This way they take turns, sending a message between them, and reversing the direction after each exchange.

3. As a first step, establish the connection, create all i/o objects. After that you will need a loop that repeatedly allows sending back and forth multiple messages. For the server this will look as follows:

```
establish the connection, get a socket
create the reader and writer objects
create a scanner object

// loop infinitely
while(loop == true)
{
      // talk
      ask for input of a string using the scanner
      send that string using the writer
      // listen
      listen to a response using the reader
      print the response
}
close the connection
```

   The client will have to revert that order, i.e. it needs a similar loop in which it first listens, then talks.
   Your programs must display meaningful prompts. One should never sit in front of a blank screen without indication what input is expected or what the program is waiting for.

4. Consider implementing both programs before you move on to the next step. The loops will run infinitely but can be stopped by manually terminating the program.

5. An infinite chat is not very useful. Modify your code such that when either participant enters the string "bye" the following happens:
   - "bye" gets sent to the other participant
   - the sender's loop breaks, and their app shuts down automatically

- the receiver sees “bye” on their screen, their loop breaks, and their app also shuts down automatically

6. Criteria:
   - commented code
   - meaningful prompts
   - correctness, completeness, stipulations are met
   - clean, properly formatted code
   - code compiles without warnings or errors
   - no runtime exceptions or crashes

7. You will demo your program and also submit the Java files for your chat client and server.