**PROJECT 1**

**CECS 326**

**REPORT**

**BY AVLOKITA SHARMA**

**ID: 025607255**

**DUE DATE: 02/26/2023**

**VIDEO: https://youtu.be/iAdpl6EJK-4**

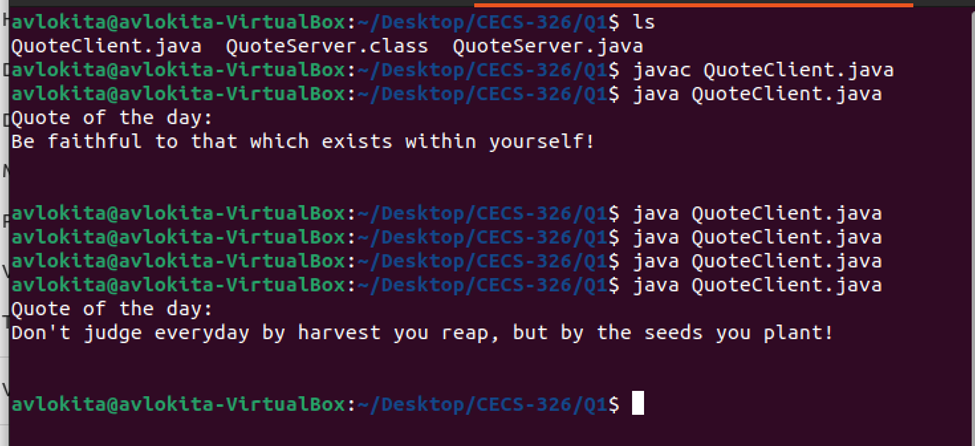
**Question 1**

For the QuoteServer.java file:

1. First, an array of String with nine quotes is created.
2. Using try and catch a new Server Socket on port 6017 is created.
3. For listening connections, a while loop is created, until the client gets connected with the server.
4. In the while loop, a socket named client is created that accepts the socket. Using PrintWriter, we are able to send bytes the to client.
5. To make a random quote from the string of array created, random\_quote is used to generate a random index from the string of arrays.
6. Then the Quote of The Day is printed, and the client-server connection is closed until ran again in the terminal.

For the QuoteClient.java file:

1. A new socket is created on port 6017. Port 6017 becomes the local host.
2. An input stream is generated to get the stream between the client and the server.
3. If a string is seen, then a variable gets created, otherwise no.
4. A bytearray is created to keep all the bytes that are coming from the stream.
5. Then all those bytes are displayed converting them to strings
6. Finally, the connection is closed until ran again in the terminal.



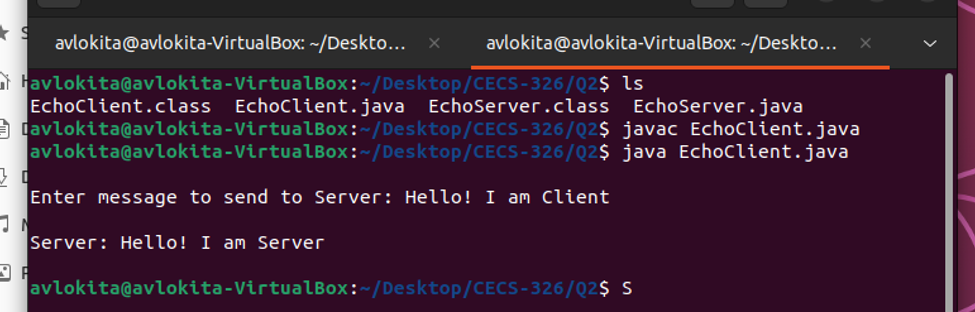
**Question 2**

For EchoServer.java:

1. A new socket is created on port 6007.
2. To get stream by the client, an input stream is created.
3. Now, to save the stream, BufferedReader is used to save the stream into the buffer. It buffers the characters to enable reading of text data.
4. Now, the stream gets converted into string and as a client, a message is asked to send to the server.
5. That message should be echoed back as well; therefore I created a string called echoback
6. If Client has messaged something then Client’s reply should be replaced with the server and echoed back to the server side. Else, it would just say that the client didn’t say anything.
7. The connection is closed until ran again in the terminal.

For EchoClient.java:

1. A socket is created on port 6007.
2. Bytes are sent to the server.
3. A scan object is created to send a message to the Server.
4. The message is printed out.
5. To catch the stream sent by the server, inst object is created.
6. BufferedReaer is used to save the stream int on the buffer.
7. Like previous steps, the stream gets converted into string using readline() method and the reply gets printed back.
8. The Connection is closed until rain again.



Text

Description automatically generated

***[If more explanation is needed, then check out my video link]***