✔ **Done:** Make a submission

Opened: Thursday, 14 March 2024, 12:05 AM
Due: Thursday, 21 March 2024, 11:55 PM

Assignment Title: Online Chat Application

This assignment aims to assess your skills in socket programming, client-server communication, and user interface design.

Assignment Instructions

You are tasked with developing a simple online chat application using Java. The application should allow multiple users to connect to a central server, send messages, and receive messages from other users.

Requirements:

1. **Server Implementation:**

   a. Create a server class, ChatServer, using socket programming to manage connections from multiple clients.
   b. The server should be able to handle incoming connections, assign a unique user ID to each connected client, and maintain a list of connected users.

2. **Client Implementation:**

   a. Implement a client class, ChatClient, that connects to the server using sockets.

   b. Each client should be able to send messages to the server, which will broadcast the messages to all connected clients.

   c. Clients should also be able to receive messages from other users.

3. **User Interface:**

   a. Include a simple text-based for the client to facilitate message input and display.

Guidelines

- Submit well-commented Java source code.
- Screenshot of the Text based User interface.
- Include a README file explaining how to run your chat application and providing details about your implementation.

Grading Criteria

1. Server Implementation: Implementation of socket programming concepts, user management, connection handling.
2. Client Implementation: Implementation of socket connection, messaging sending, Message broadcasting
3. User Interface: Creation of a user interface which is usable and provides screenshots of output of the interface.
4. Logic and Computation
5. Program Flow and Structure
6. Code style and readability.

?

Submission Instructions

- Read the rubric on how you are going to be graded before you start to work on this assignment.
- Remember to use appropriate variable names and follow best practices of coding. Please provide a screenshot of the outputs. Submit the assignment in MS Word or PDF file.

This assignment will be assessed by your instructor using the rubric below.

Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading
Grading status	Graded
Time remaining	Assignment was submitted 1 day 14 hours early
Last modified	Wednesday, 20 March 2024, 9:12 AM
File submissions	Programming Assignment Unit 7.pdf 20 March 2024, 9:12 AM
Submission comments	Comments (0)

Grading criteria

Server Implementation	ChatServer class created to manage connections from multiple clients. Implemented code that is able to handle incoming connections, assign a unique user ID to each connected client, and maintain a list of connected users and connection handling efficiently. ***20 points***	ChatServer class created to manage connections from multiple clients. However, user management and connection handling have not been implemented correctly. ***16 points***	ChatServer class is created but it is not able to manage multiple clients. User management and Connection handling have also not been implemented correctly. ***10 points***	Incorrect implementation of Server ***0 points***

Client Implementation	Implemented client class, ChatClient, that connects to the server using sockets. On this implementation, the client will be able to send messages to the server, which will then broadcast the messages to all connected clients. Clients will also be able to receive messages from other users. ***20 points***	Implemented client class, ChatClient, that connects to the server using sockets. On this implementation, the client will be able to send messages to the server, but the server will not be able to broadcast the messages to all connected clients. Clients will not be able to receive the broadcast message. ***10 points***	Incorrect implementation of client. ***0 points***	
User Interface	Provided screen shot of a simple text-based user interface for the client to facilitate message input and display. ***10 points***	The program does not provide a screenshot of the program's text-based user interface. ***0 points***		
Logic and Computation	The program accurately demonstrated the operations of Online Chat Application. ***20 points***	The program accurately demonstrates some of the operations of the Online Chat Application ***10 points***	The program does not correctly demonstrate the operations of Online Chat Application ***0 points***	
Program Flow and Structure	The program follows a logical flow and is well-structured. Proper variable declaration and initialization are done. Meaningful variable names and appropriate data types are used. ***20 points***	The program follows a logical flow and is structured. Variable names and data types used are not correct. ***10 points***	Variables are not declared, no proper logical flow and inappropriate data types declared. ***0 points***	
Code Style and Readability	The code follows consistent indentation and formatting conventions. The code is easy to read and understand. The program does not contain any unnecessary or redundant code. ***10 points***	The code does not have proper indentation and formatting. ***8 points***	Unnecessary code and no proper indentation are followed. ***5 points***	Redundant code. ***0 points***

Feedback

Grade	4.50 / 10.00
Graded on	Wednesday, 20 March 2024, 11:08 AM
Graded by	 Vikas Thada (Instructor)

Feedback comments

Dear Godfrey

Exactly copied from ChatGPT with minor changes on message.

Regards

Grade breakdown

Server Implementation	ChatServer class created to manage connections from multiple clients. Implemented code that is able to handle incoming connections, assign a unique user ID to each connected client, and maintain a list of connected users and connection handling efficiently. ***20 points***	ChatServer class created to manage connections from multiple clients. However, user management and connection handling have not been implemented correctly. ***16 points***	ChatServer class is created but it is not able to manage multiple clients. User management and Connection handling have also not been implemented correctly. ***10 points***	Incorrect implementation of Server ***0 points***
Client Implementation	Implemented client class, ChatClient, that connects to the server using sockets. On this implementation, the client will be able to send messages to the server, which will then broadcast the messages to all connected clients. Clients will also be able to receive messages from other users. ***20 points***	Implemented client class, ChatClient, that connects to the server using sockets. On this implementation, the client will be able to send messages to the server, but the server will not be able to broadcast the messages to all connected clients. Clients will not be able to receive the broadcast message. ***10 points***	Incorrect implementation of client. ***0 points***	
User Interface	Provided screen shot of a simple text-based user interface for the client to facilitate message input and display. ***10 points***		The program does not provide a screenshot of the program's text-based user interface. ***0 points***	
Logic and Computation	The program accurately demonstrated the operations of Online Chat Application. ***20 points***	The program accurately demonstrates some of the operations of the Online Chat Application ***10 points***	The program does not correctly demonstrate the operations of Online Chat Application ***0 points***	

Program Flow and Structure	The program follows a logical flow and is well-structured. Proper variable declaration and initialization are done. Meaningful variable names and appropriate data types are used. ***20 points***	The program follows a logical flow and is structured. Variable names and data types used are not correct. ***10 points***	Variables are not declared, no proper logical flow and inappropriate data types declared. ***0 points***	
Code Style and Readability	The code follows consistent indentation and formatting conventions. The code is easy to read and understand. The program does not contain any unnecessary or redundant code. ***10 points***	The code does not have proper indentation and formatting. ***8 points***	Unnecessary code and no proper indentation are followed. ***5 points***	Redundant code. ***0 points***