coding problems using socket module

1. **Echo Server:** Description: Create a simple server using Python's **socket** module that echoes back any message received from a client. Upon connection, the server should accept messages from the client and send the same message back to the client. This is a fundamental exercise to understand the basic communication between a client and a server using sockets.
2. **Guessing Game:** Description: Develop a server-client guessing game using Python's **socket** module. The server generates a random number, and the client tries to guess it by sending guesses to the server. The server provides hints (higher or lower) until the client guesses the correct number. This exercise helps in understanding how to handle communication between client and server for interactive games.
3. **Chat Application:** Description: Implement a basic chat application using Python's **socket** module where multiple clients can connect to a server and communicate with each other. Clients should be able to send messages to the server, which will broadcast them to all other connected clients. Additional features like joining and leaving notifications can be included to enhance the functionality.
4. **File Transfer Client:** Description: Design a simple file transfer client using Python's **socket** module. The client connects to a server and requests a file to download. The server sends the requested file to the client, which saves it locally. Ensure error handling for scenarios such as file not found or server connection issues. This exercise helps in understanding how to transfer files over a network using sockets.
5. **Simple Calculator Server:** Description: Create a server-client application using Python's **socket** module where clients can send simple arithmetic expressions (e.g., "2 + 3") to the server. The server evaluates these expressions and sends back the result to the client. This exercise helps in understanding how to process and evaluate data received from clients and send responses accordingly.