

1. Implement chatting application using java socket programming

—>

Server Code:

```
import java.io.*;
import java.net.*;
import java.util.*;

public class ChatServer
{
    private static final int PORT = 12345;
    private static Set<PrintWriter> clientWriters = new HashSet<>();

    public static void main(String[] args) throws IOException
    {
        System.out.println("Chat server started...");
        ServerSocket serverSocket = new ServerSocket(PORT);

        while (true) {
            Socket clientSocket = serverSocket.accept();
            System.out.println("Client connected: " + clientSocket);
            new ClientHandler(clientSocket).start();
        }
    }

    private static class ClientHandler extends Thread
    {
        private Socket socket;
        private PrintWriter out;
        private BufferedReader in;

        public ClientHandler(Socket socket) {
            this.socket = socket;
        }

        public void run()
        {
            try {
                in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
                out = new PrintWriter(socket.getOutputStream(), true);

                synchronized (clientWriters) {
                    clientWriters.add(out);
                }

                String message;
                while ((message = in.readLine()) != null) {
                    System.out.println("Received: " + message);
                    synchronized (clientWriters) {
                        for (PrintWriter writer : clientWriters) {
                            writer.println(message);
                        }
                    }
                }
            }
            catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

        }
    }
}
} catch (IOException e) {
    System.out.println("Error handling client: " + e.getMessage());
} finally {
    try {
        socket.close();
    } catch (IOException e) {}
    synchronized (clientWriters) {
        clientWriters.remove(out);
    }
}
}
}
}
}

```

Client Code:

```

import java.io.*;
import java.net.*;

public class ChatClient
{
    private static final String SERVER_ADDRESS = "localhost";
    private static final int PORT = 12345;

    public static void main(String[] args) {
        try (
            Socket socket = new Socket(SERVER_ADDRESS, PORT);
            BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        {
            System.out.println("Connected to chat server");

            new Thread(() -> {
                String response;
                try {
                    while ((response = in.readLine()) != null) {
                        System.out.println(response);
                    }
                } catch (IOException e) {
                    System.out.println("Disconnected from server.");
                }
            }).start();

            String userInput;
            while ((userInput = input.readLine()) != null) {
                out.println(userInput);
            }
        }
    }
}

```

```

    }

    } catch (IOException e) {
        System.err.println("Error: " + e.getMessage());
    }
}
}

```

Server Output:

```

Chat server started...
Client connected: Socket[addr=/
127.0.0.1,port=54321,localport=12345]
Client connected: Socket[addr=/
127.0.0.1,port=54322,localport=12345]
Received: Virendra: Hello!
Received: Prajval: Hi Virendra!

```

Client Output:

```

Connected to chat server
Virendra: Hello!
Prajval: Hi Virendra!

```

2. Develop a TCP client-server application where the client sends two numbers to the server, and the server returns their sum.

—>

Server Code:

```

import java.io.*;
import java.net.*;

public class SumServer
{
    private static final int PORT = 12345;

    public static void main(String[] args)
    {
        try (ServerSocket serverSocket = new ServerSocket(PORT)) {
            System.out.println("Server started. Waiting for client...");

            Socket socket = serverSocket.accept();
            System.out.println("Client connected: " + socket);

            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            String num1Str = in.readLine();
            String num2Str = in.readLine();

            int num1 = Integer.parseInt(num1Str);
            int num2 = Integer.parseInt(num2Str);
            int sum = num1 + num2;

```

Server Output:

```

Server started. Waiting for client...
Client connected: Socket[addr=/
127.0.0.1,port=xxxx,localport=12345]
Connection closed.

```

```

        out.println("Sum is: " + sum);

        socket.close();
        System.out.println("Connection closed.");
    } catch (IOException e) {
        System.out.println("Server error: " + e.getMessage());
    }
}
}
}

```

Client Code:

```

import java.io.*;
import java.net.*;

public class SumClient
{
    private static final String SERVER_ADDRESS = "localhost";
    private static final int PORT = 12345;

    public static void main(String[] args)
    {
        try (
            Socket socket = new Socket(SERVER_ADDRESS, PORT);
            BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        ) {
            System.out.print("Enter first number: ");
            String num1 = input.readLine();

            System.out.print("Enter second number: ");
            String num2 = input.readLine();

            out.println(num1);
            out.println(num2);

            String response = in.readLine();
            System.out.println("Server response: " + response);

        } catch (IOException e) {
            System.out.println("Client error: " + e.getMessage());
        }
    }
}

```

Client Output:

```

Enter first number: 10
Enter second number: 25
Server response: Sum is: 35

```