Task: **Socket programing**

- Write a client-server multi-threaded program to convert an input text string to Upper Case.

Codes:

Server side:

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

           // Send a welcome message to the client.
           out.println("Hello, you are client #" + clientNumber + ".");
           out.println("Enter a line with only a period to quit\n");

           // Get messages from the client, line by line; return them
           // capitalized
           while (true) {
              String input = in.readLine();
              if (input == null || input.equals(".")) {
                 break;
              }
              out.println(input.toUpperCase());
           }
        } catch (IOException e) {
           log("Error handling client# " + clientNumber + ": " + e);
        } finally {
           try {
              socket.close();
           } catch (IOException e) {
              log("Couldn't close a socket, what's going on?");
           }
           log("Connection with client# " + clientNumber + " closed");
        }
     }
```

```
public static void main(String[] args) throws Exception {
      System.out.println("The capitalization server is running.");
      int clientNumber = 0;
      ServerSocket listener = new ServerSocket(9898);
      try {
         while (true) {
            new Capitalizer(listener.accept(), clientNumber++).start();
         }
      } finally {
         listener.close();
      }
   }

// ------------ the variables and constructor ------------------------

private Socket socket;
      private int clientNumber;

      public Capitalizer(Socket socket, int clientNumber) {
         this.socket = socket;
         this.clientNumber = clientNumber;
         log("New connection with client# " + clientNumber + " at " + socket);
      }
// ------------------- log() method --------------------
private void log(String message) {
         System.out.println(message);
      }
```

Client side:

```
// Layout GUI
      messageArea.setEditable(false);
      frame.getContentPane().add(dataField, "North");
      frame.getContentPane().add(new JScrollPane(messageArea), "Center");

      // Add Listeners
      dataField.addActionListener(new ActionListener() {
         /**
          * Responds to pressing the enter key in the textfield
          * by sending the contents of the text field to the
          * server and displaying the response from the server
          * in the text area.  If the response is "." we exit
          * the whole application, which closes all sockets,
          * streams and windows.
          */
         public void actionPerformed(ActionEvent e) {
            out.println(dataField.getText());
              String response;
            try {
               response = in.readLine();
               if (response == null || response.equals("")) {
                    System.exit(0);
                 }
            } catch (IOException ex) {
                 response = "Error: " + ex;
            }
            messageArea.append(response + "\n");
            dataField.selectAll();
         }
      });
   }
```

```
public void connectToServer() throws IOException {

    // Get the server address from a dialog box.
    String serverAddress = JOptionPane.showInputDialog(
      frame,
      "Enter IP Address of the Server:",
      "Welcome to the Capitalization Program",
      JOptionPane.QUESTION_MESSAGE);

    // Make connection and initialize streams
    Socket socket = new Socket(serverAddress, 9898);
    in = new BufferedReader(
          new InputStreamReader(socket.getInputStream()));
    out = new PrintWriter(socket.getOutputStream(), true);

    // Consume the initial welcoming messages from the server
    for (int i = 0; i < 3; i++) {
      messageArea.append(in.readLine() + "\n");
    }
  }
```