Program Code:-

Write a program using TCP socket for wired network for following

- **Say Hello to Each other**

1. HelloServer.java

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

public class HelloServer {
    public static void main(String[] args) {
        try (ServerSocket serverSocket = new ServerSocket(12345)) {
            System.out.println("Server is listening on port 12345");

            while (true) {
                Socket socket = serverSocket.accept();
                System.out.println("New client connected");

                new ClientHandler(socket).start();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

class ClientHandler extends Thread {
    private Socket socket;
    public ClientHandler(Socket socket) {
        this.socket = socket;
    }
    public void run() {
        try (BufferedReader input = new BufferedReader(new
      InputStreamReader(socket.getInputStream()));
                PrintWriter output = new PrintWriter(socket.getOutputStream(), true)) {

            String clientMessage = input.readLine();
            System.out.println("Received from client: " + clientMessage);
            output.println("Hello, Client!");

        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            try {
                socket.close();
            } catch (IOException ex) {
                ex.printStackTrace();
            }
            }
        }
    }
```

2. HelloClient.java

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

public class HelloClient {
    public static void main(String[] args) {
        try (Socket socket = new Socket("localhost", 12345);
                PrintWriter output = new PrintWriter(socket.getOutputStream(), true);
                BufferedReader input = new BufferedReader(new
InputStreamReader(socket.getInputStream()))) {

            output.println("Hello, Server!");
            String response = input.readLine();
            System.out.println("Server says: " + response);

        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

OUTPUT:

```
D:\GITHUB\LAB\5TH SEMESTER\CNS\Assign 9>java HelloServer
Server is listening on port 12345
New Client connected
Received from client: Hello, Server!

D:\GITHUB\LAB\5TH SEMESTER\CNS\Assign 9>java HelloClient
Server says: Hello, Client!
```

- **File transfer**
  1. FileTransferServer.java

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

public class FileTransferServer {
    public static void main(String[] args) {
        try (ServerSocket serverSocket = new ServerSocket(12346)) {
            System.out.println("File Transfer Server is listening on port 12346");

            while (true) {
                Socket socket = serverSocket.accept();
                System.out.println("Client connected");

                new FileTransferHandler(socket).start();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

class FileTransferHandler extends Thread {
    private Socket socket;

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

    public void run() {
        try (DataInputStream input = new DataInputStream(socket.getInputStream());
                FileOutputStream fileOutput = new
FileOutputStream("received_file.txt")) {

            int bytesRead;
            byte[] buffer = new byte[4096];

            while ((bytesRead = input.read(buffer)) != -1) {
                fileOutput.write(buffer, 0, bytesRead);
            }

            System.out.println("File received!");

        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            try {
                socket.close();
            } catch (IOException ex) {
                ex.printStackTrace();
            }
        }
    }
}
```

2. FileTransferClient.java

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

public class FileTransferClient {
    public static void main(String[] args) {
        try (Socket socket = new Socket("localhost", 12346);
                FileInputStream fileInput = new FileInputStream("file_to_send.txt");
                DataOutputStream output = new
DataOutputStream(socket.getOutputStream())) {

            byte[] buffer = new byte[4096];
            int bytesRead;

            while ((bytesRead = fileInput.read(buffer)) != -1) {
                output.write(buffer, 0, bytesRead);
            }
            System.out.println("File sent!");
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

Outuput:

```
D:\GITHUB\LAB\5TH SEMESTER\CNS\Assign 9>java FileTransferServer
File Transfer Server is listening on port 12346
Client connected
File received!

D:\GITHUB\LAB\5TH SEMESTER\CNS\Assign 9>java FileTransferClient
File sent!
```

- **Calculator**

1. CalculatorServer.java

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

public class CalculatorServer {
    public static void main(String[] args) {
        try (ServerSocket serverSocket = new ServerSocket(12347)) {
            System.out.println("Calculator Server is listening on port 12347");

            while (true) {
                Socket socket = serverSocket.accept();
                System.out.println("Client connected");

                new CalculatorHandler(socket).start();
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}

class CalculatorHandler extends Thread {
    private Socket socket;

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

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

            String clientInput;
            while ((clientInput = input.readLine()) != null) {
                String[] parts = clientInput.split(" ");
                double num1 = Double.parseDouble(parts[0]);
                String operator = parts[1];
                double num2 = Double.parseDouble(parts[2]);
                double result = 0;

                switch (operator) {
                    case "+":
                        result = num1 + num2;
                        break;
                    case "-":
                        result = num1 - num2;
                        break;
                    case "*":
                        result = num1 * num2;
                        break;
                    case "/":
                        if (num2 != 0) {
                            result = num1 / num2;
                        } else {
                            output.println("Error: Division by zero");
                            continue;
                        }
                        break;
                    default:
                        output.println("Error: Invalid operator");
                        continue;
                }

                output.println("Result: " + result);
            }

        } catch (IOException ex) {
            ex.printStackTrace();
        } finally {
            try {
                socket.close();
            } catch (IOException ex) {
                ex.printStackTrace();
            }
        }
    }
}
```

2. CalculatorClient.java

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

public class CalculatorClient {
    public static void main(String[] args) {
        try (Socket socket = new Socket("localhost", 12347);
                PrintWriter output = new PrintWriter(socket.getOutputStream(), true);
                BufferedReader input = new BufferedReader(new
InputStreamReader(socket.getInputStream()))) {

            BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));
```

```
            String expression;

            System.out.println("Enter calculations in the format 'num1 operator num2' (e.g., '3 + 4'): ");
            while ((expression = userInput.readLine()) != null) {
                output.println(expression);
                String response = input.readLine();
                System.out.println(response);
            }

        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
}
```

OUTPUT:

```
D:\GITHUB\LAB\5TH SEMESTER\CNS\Assign 8>java CalculatorServer

Calculator Server is listening on port 12347

Client connected
```

```
D:\GITHUB\LAB\5TH SEMESTER\CNS\Assign 8>java CalculatorClient

Enter calculations in the format 'num1 operator num2' (e.g., '3 + 4'):

3 * 4

Result: 12.0

12 + 23

Result: 35.0

25 / 2

Result: 12.5

25 - 4

Result: 21.0
```