

ASSIGNMENT 9

Experiment

IMPLEMENTING THE CONCEPT OF SOCKET PROGRAMMING

Code: (Server)

```
import java.net.*;
import java.io.*;
public class Server {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = null;
        try {
            serverSocket = new ServerSocket(8085);
        } catch (IOException e) {
            System.err.println("Could not listen on port: 8080.");
            System.exit(1);
        }
        Socket clientSocket = null;
        try {
            System.out.println("Waiting for connection...");
            clientSocket = serverSocket.accept();
            System.out.println("Connection established with " +
                clientSocket.getInetAddress().getHostName() + ":" + clientSocket.getPort());
        } catch (IOException e) {
            System.err.println("Accept failed.");
            System.exit(1);
        }
        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new
            InputStreamReader(clientSocket.getInputStream()));
        String inputLine;
        while ((inputLine = in.readLine()) != null) {
            System.out.println("Received: " + inputLine);
            out.println("Echo: " + inputLine);
            if (inputLine.equals("quit"))
                break;
        }
        out.close();
        in.close();
        clientSocket.close();
        serverSocket.close();
    }
}
```

Code: (Client)

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

```

public class Client {
    public static void main(String[] args) throws IOException {
        Socket socket = null;
        try {
            socket = new Socket("localhost", 8085);
        } catch (UnknownHostException e) {
            System.err.println("Don't know about host: localhost.");
            System.exit(1);
        } catch (IOException e) {
            System.err.println("Couldn't get I/O for the connection to:
localhost.");
            System.exit(1);
        }
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
        BufferedReader stdIn = new BufferedReader(new
InputStreamReader(System.in));
        String userInput;
        while ((userInput = stdIn.readLine()) != null) {
            out.println(userInput);
            System.out.println("Server says: " + in.readLine());
            if (userInput.equals("quit"))
                break;
        }
        out.close();
        in.close();
        stdIn.close();
        socket.close();
    }
}

```

Output

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 JUPYTER
abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07$ javac Server.java
abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07$ java Server
Waiting for connection...
Connection established with localhost:50452
Received: hello
Received: Sally went to the seashore to buy seashells

```

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 JUPYTER
abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07$ javac Client.java
abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07$ java Client
hello
Server says: Echo: hello
Sally went to the seashore to buy seashells
Server says: Echo: Sally went to the seashore to buy seashells

```

EXPERIMENT :

Using socket programming find the factorial of a number which is generated by the client

Code: (Server)

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

public class fac_server {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = null;
        try {
            serverSocket = new ServerSocket(8085);
            System.out.println("Waiting for client connection...");
        } catch (IOException e) {
            System.err.println("Could not listen on port: 9999.");
            System.exit(1);
        }
        while (true) {
            Socket clientSocket = null;
            try {
                clientSocket = serverSocket.accept();
                System.out.println("Connection received from: " +
                    clientSocket.getInetAddress().getHostName());
            } catch (IOException e) {
                System.err.println("Accept failed.");
                System.exit(1);
            }
            BufferedReader in = new BufferedReader(new
                InputStreamReader(clientSocket.getInputStream()));
            String number = in.readLine();
            System.out.println("Data Received: " + number);
            int n = Integer.parseInt(number);
            int result = factorial(n);
            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(),
                true);
            out.println(result);
            out.close();
            in.close();
            clientSocket.close();
            serverSocket.close();
        }
    }

    public static int factorial(int n) {
```

```

if (n == 0) {
return 1;
} else {
return n * factorial(n - 1);
}
}
}
Code: (Client)
import java.net.*;
import java.io.*;
public class Client {
public static void main(String[] args) throws IOException {
Socket socket = null;
try {
socket = new Socket("localhost", 8085);
} catch (UnknownHostException e) {
System.err.println("Don't know about host: localhost.");
System.exit(1);
} catch (IOException e) {
System.err.println("Couldn't get I/O for the connection to:
localhost.");
System.exit(1);
}
PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
BufferedReader stdIn = new BufferedReader(new
InputStreamReader(System.in));
String userInput;
while ((userInput = stdIn.readLine()) != null) {
out.println(userInput);
System.out.println("Server says: " + in.readLine());
if (userInput.equals("quit"))
break;
}
out.close();
in.close();
stdIn.close();
socket.close();
}
}

```

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER

● abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07/Program 02$ javac fac_server.java
● abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07/Program 02$ java fac_server
Waiting for client connection...
Connection received from: localhost
Data Received: 5
Accept failed.
○ abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07/Program 02$ █
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  JUPYTER

● abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07/Program 02$ javac fac_client.java
● abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07/Program 02$ java fac_client
Enter a number: 5
The factorial of 5 is 120
○ abc@7a04e2b33951:/data/Computer Network Lab/Assignment - 07/Program 02$ █
```