

Practical 11:- Networking

1. Implement Echo client/server program using TCP

(i) server

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

public class EchoServer {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(5000);
        System.out.println("Echo Server started on port 5000");

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

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

            String received;
            while ((received = in.readLine()) != null) {
                System.out.println("Received from client: " + received);
                out.println("Echo: " + received);
            }

            clientSocket.close();
        }
    }
}
```

(ii) client

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

public class EchoClient {
    public static void main(String[] args) throws IOException {
        Socket socket = new Socket("localhost", 5000);
```

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

String userInput;
System.out.println("Enter message to send (type 'exit' to quit):");

while (!(userInput = input.readLine()).equalsIgnoreCase("exit")) {
out.println(userInput);
    System.out.println("Server response: " + in.readLine());
}

socket.close();
}
```

Output :-

```
PS C:\12302130501036> javac EchoServer.java
PS C:\12302130501036> java EchoServer
Echo Server started on port 5000
Client connected: /127.0.0.1
Received from client: hello
PS C:\12302130501036> javac EchoClient.java
PS C:\12302130501036> java EchoClient
Enter message to send (type 'exit' to quit):
hello
Server response: Echo: hello
exit
```

2. Write a program using UDP which give name of the audio file to server and server reply with content of audio file

(i) server

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

public class AudioFileServer {
    public static void main(String[] args) throws IOException {
        DatagramSocket socket = new DatagramSocket(6000);    byte[]
        buffer = new byte[65536];
```

```
        System.out.println("UDP Audio Server is running...");
```

```

while (true) {
    DatagramPacket request = new DatagramPacket(buffer, buffer.length);
    socket.receive(request);

    String filename = new String(request.getData(), 0, request.getLength());
    System.out.println("Client requested: " + filename);

    File file = new File(filename);
    String response;

    if (file.exists() && file.isFile()) {
        BufferedReader reader = new BufferedReader(new FileReader(file));
        StringBuilder content = new StringBuilder();
        String line;
        while ((line = reader.readLine()) != null) {
            content.append(line).append("\n");
        }
        reader.close();
        response = content.toString();
    } else {
        response = "File not found.";
    }

    byte[] responseData = response.getBytes();
    DatagramPacket responsePacket = new DatagramPacket(
responseData,          responseData.length,
request.getAddress(),   request.getPort()
    );
    socket.send(responsePacket);
}
}

```

(ii)client

```

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

public class AudioFileClient {
    public static void main(String[] args) throws IOException {
        DatagramSocket socket = new DatagramSocket();
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Enter the audio filename: ");
        String filename = reader.readLine();
    }
}

```

```
byte[] buffer = filename.getBytes();
InetAddress serverAddress = InetAddress.getByName("localhost");

DatagramPacket request = new DatagramPacket(buffer, buffer.length, serverAddress,
6000);
socket.send(request);

byte[] responseBuffer = new byte[65536];
DatagramPacket response = new DatagramPacket(responseBuffer, responseBuffer.length);
socket.receive(response);

String responseText = new String(response.getData(), 0, response.getLength());
System.out.println("Server Response:\n" + responseText);
}
}
```

Output :-

```
PS C:\12302130501036> javac AudioFileServer.java
PS C:\12302130501036> java AudioFileServer
UDP Audio Server is running...
Client requested: song.txt
PS C:\12302130501036> javac AudioFileClient.java
PS C:\12302130501036> java AudioFileClient
Enter the audio filename: song.txt
Server Response:
File not found.
```