**Practical - 3**

**Aim: Implement IPC with sockets**

**Code:**

***Client.java***

import java.net.\*;

import java.io.\*;

public class Client {

private Socket socket = null;

private DataInputStream input = null;

private DataOutputStream output = null;

public Client(String address, int port) {

try {

socket = new Socket(address, port);

System.out.println("Connected");

input = new DataInputStream(System.in);

output = new DataOutputStream(socket.getOutputStream());

} catch (UnknownHostException u) {

System.out.println(u);

} catch (IOException i) {

System.out.println(i);

}

String line = "";

while (!line.equals("Over")) {

try {

line = input.readLine();

output.writeUTF(line);

} catch (IOException i) {

System.out.println(i);

}

}

try {

input.close();

output.close();

socket.close();

} catch (IOException i) {

System.out.println(i);

}

}

public static void main(String args[]) {

Client client = new Client("127.0.0.1", 5000);

}

}

***Server.java***

import java.io.\*;

import java.net.\*;

public class Server {

private Socket socket = null;

private ServerSocket server = null;

private DataInputStream input = null;

public Server(int port) {

try {

server = new ServerSocket(port);

System.out.println("Server started");

System.out.println("Waiting for a client ...");

socket = server.accept();

System.out.println("Client accepted");

input = new DataInputStream(new BufferedInputStream(socket.getInputStream()));

String line = "";

while (!line.equals("Over")) {

try {

line = input.readUTF();

System.out.println(line);

} catch (IOException i) {

System.out.println(i);

}

}

System.out.println("Closing connection");

socket.close();

input.close();

} catch (IOException i) {

System.out.println(i);

}

}

public static void main(String args[]) {

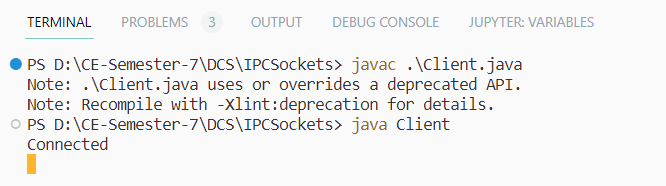
Server server = new Server(5000);

}

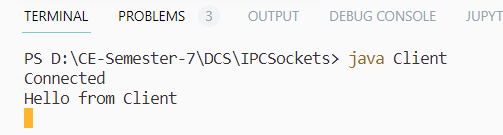
}

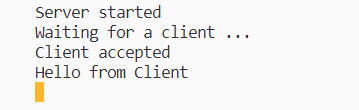
**Output:**

****

****

****

****

****