**Exp No: 7 PROGRAM FOR FILE TRANSFER**

**Date:**

**AIM**

**ALGORITHM**

**CODE**

**Server.java**

import java.io.\*;

import java.net.\*;

public class Server {

private static DataInputStream dataInputStream=null;

private static DataOutputStream dataOutputStream=null;

public static void main(String[] args) throws IllegalArgumentException {

if (args.length != 1) {

throw new IllegalArgumentException("One argument is required. Path to store the recieving file (exclude format).");

}

try {

ServerSocket serverSocket = new ServerSocket(900);

Socket socket = serverSocket.accept();

System.out.println("Connection Established");

dataInputStream= new DataInputStream(socket.getInputStream());

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

receiveFile(args[0]);

dataInputStream.close();

dataOutputStream.close();

socket.close();

} catch (Exception e) {

e.printStackTrace();

}

}

private static void receiveFile(String path) throws Exception {

int bytesRead;

String file\_format = dataInputStream.readUTF();

FileOutputStream fileOutputStream = new FileOutputStream(path + "." + file\_format);

long sizeLeft = dataInputStream.readLong();

byte[] buffer = new byte[4 \* 1024];

do {

bytesRead = dataInputStream.read(buffer, 0, (int) Math.min(buffer.length, sizeLeft));

fileOutputStream.write(buffer);

sizeLeft -= bytesRead; // subtracting bytes read from size

} while (sizeLeft > 0 && bytesRead != -1);

System.out.println("File is Received and Stored at " + path + "." + file\_format);

fileOutputStream.close();

}

}

**Client.java**

import java.io.\*;

import java.net.Socket;

public class Client {

private static DataOutputStream dataOutputStream = null;

private static DataInputStream dataInputStream = null;

public static void main(String[] args) throws IllegalArgumentException {

if (args.length != 1) {

throw new IllegalArgumentException("One argument is required. Path of the file to be sent.");

}

try {

Socket socket = new Socket("localhost",900);

dataInputStream = new DataInputStream(socket.getInputStream());

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

System.out.println("Sending the File to the Server");

sendFile(args[0]);

dataInputStream.close();

dataInputStream.close();

} catch (Exception e) {

e.printStackTrace();

}

}

private static void sendFile(String path) throws Exception {

int bytesRead;

File file = new File(path);

String[] splittedStr = path.split("\\.", 0);

dataOutputStream.writeUTF(splittedStr[splittedStr.length - 1]);

FileInputStream fileInputStream = new FileInputStream(file);

int sizeLeft = (int) file.length();

dataOutputStream.writeLong(sizeLeft);

byte[] buffer = new byte[4 \* 1024];

do {

bytesRead = fileInputStream.read(buffer, 0, (int) Math.min(buffer.length, sizeLeft));

dataOutputStream.write(buffer, 0, bytesRead);

dataOutputStream.flush();

sizeLeft -= bytesRead; // subtracting bytes read from size

} while (sizeLeft > 0 && bytesRead != -1);

fileInputStream.close();

dataOutputStream.close();

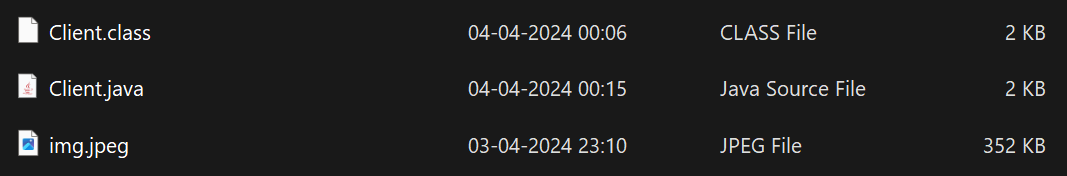
}

}

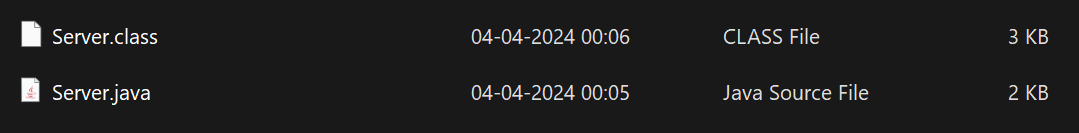
**OUTPUT**

*Before:*

**Client:**

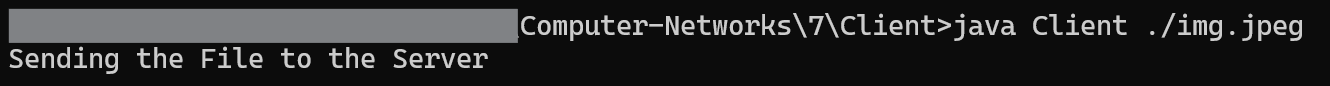
****

**Server:**

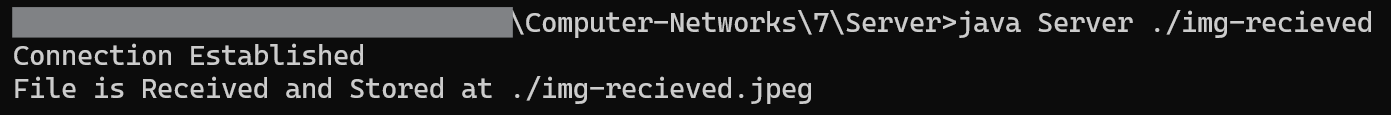
****

*Executing the program:*

**Client:**

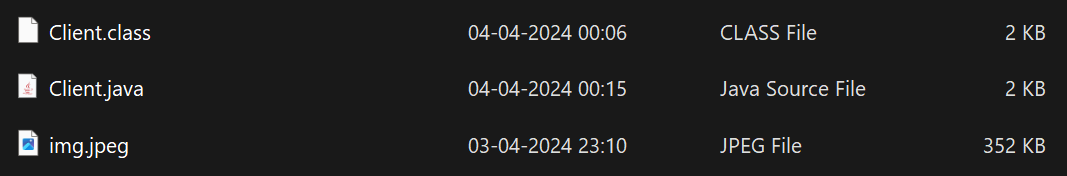


**Server:**

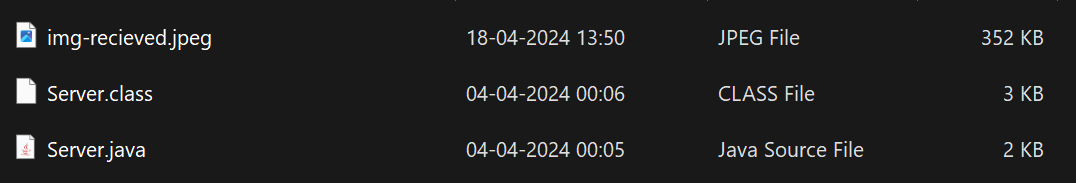
****

*After:*

**Client:**

****

**Server:**

****

**RESULT**