Client.java

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class Client
{
    public static void main(String[] args) throws IOException
    {
        try
        {
            Scanner scn = new Scanner(System.in);
            InetAddress ip = InetAddress.getByName("localhost");
            Socket s = new Socket(ip, 5056);
            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
            while (true)
            {
                System.out.println(dis.readUTF());
                String tosend = scn.nextLine();
                dos.writeUTF(tosend);
                if(tosend.equals("Exit"))
                {
                    System.out.println("Closing this connection : " + s);
                    s.close();
                    System.out.println("Connection closed");
                    break;
                }
                String received = dis.readUTF();
                System.out.println(received);
            }
            scn.close();
```

```
                dis.close();

                dos.close();

            }catch(Exception e){

                e.printStackTrace();

            }

        }

}
```

Server.java

```
import java.io.*;

import java.text.*;

import java.util.*;

import java.net.*;

public class Server

{

    public static void main(String[] args) throws IOException

    {

        ServerSocket ss = new ServerSocket(5056);

        while (true)

        {

            Socket s = null;

            try

            {

                s = ss.accept();

                System.out.println("A new client is connected : " + s);

                DataInputStream dis = new DataInputStream(s.getInputStream());

                DataOutputStream dos = new
DataOutputStream(s.getOutputStream());


                System.out.println("Assigning new thread for this client");
```

```
                        Thread t = new ClientHandler(s, dis, dos);
                        t.start();
                    }
                    catch (Exception e){
                        s.close();
                        e.printStackTrace();
                    }
                }
            }
}
class ClientHandler extends Thread
{
    DateFormat fordate = new SimpleDateFormat("yyyy/MM/dd");
    DateFormat fortime = new SimpleDateFormat("hh:mm:ss");
    final DataInputStream dis;
    final DataOutputStream dos;
    final Socket s;
    public ClientHandler(Socket s, DataInputStream dis, DataOutputStream dos)
    {
        this.s = s;
        this.dis = dis;
        this.dos = dos;
    }
    @Override
    public void run()
    {
        String received;
        String toreturn;
        while (true)
        {
            try {
```

```
        dos.writeUTF("What do you want?[Date | Time]..\n"+
                    "Type Exit to terminate connection.");
        received = dis.readUTF();
        if(received.equals("Exit"))
        {
            System.out.println("Client " + this.s + " sends exit...");
            System.out.println("Closing this connection.");
            this.s.close();
            System.out.println("Connection closed");
            break;
        }
        Date date = new Date();
        switch (received) {
            case "Date" :
                toreturn = fordate.format(date);
                dos.writeUTF(toreturn);
                break;
            case "Time" :
                toreturn = fortime.format(date);
                dos.writeUTF(toreturn);
                break;
            default:
                dos.writeUTF("Invalid input");
                break;
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
try
{
```

```
            this.dis.close();

            this.dos.close();


        }catch(IOException e){

            e.printStackTrace();

        }

    }

}
```