PTCPEchoServer.java

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

public class PTCPEchoServer {
    public final static int defaultPort = 2019;

    public static void main(String args[]) {
        try {
            ServerSocket ss = new ServerSocket(defaultPort);
            System.out.println("server socket is running");
            while (true) {
                try {
                    Socket s = ss.accept();
                    // Tao xu ly
                    RequestProcessing rq = new RequestProcessing(s);
                    rq.start();
                } catch(IOException e) {
                    System.out.println("connection Error: " + e);
                }
            }
        } catch (Exception e) {
            System.out.println("Creat Socket Error: " + e);
        }
    }
}
```

RequestProcessing.java

```
import java.io.*;
import java.net.*;
class RequestProcessing extends Thread
{
    private Socket s;
    public RequestProcessing(Socket s1) {
        s = s1;
    }
    public void run() {
        try {
            OutputStream os = s.getOutputStream();
            InputStream is = s.getInputStream();
            int ch = 0;
            while(true) {
                ch = is.read();
                if(ch == -1) break;
                os.write(ch);
            }
            s.close();
        }
        catch (IOException e) {
            System.err.println("Processing Error: " + e);
        }
    }
}
```

STCPEchoServer.java

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

public class STCPEchoServer {
    public final static int defaultPort = 8080;

    public static void main(String args[]) {
        try {
            ServerSocket ss = new ServerSocket(8080);
            System.out.println("server socket is running");
            while (true) {
                Socket s = ss.accept();
                OutputStream os = s.getOutputStream();
                InputStream is = s.getInputStream();
                int ch = 0;
                while (true) {
                    ch = is.read();
                    if (ch == -1)
                    break;
                    System.out.print((char) ch);
                    os.write(ch);
                }
                s.close();
            }
        } catch (Exception e) {
            System.out.print(e.toString());
        }
    }
}
```

TCPEchoClient.java

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

public class TCPEchoClient {
        public static void main(String args[]) {
                try {
                        Socket s = new Socket(args[0], Integer.parseInt(args[1]));
                        InputStream is = s.getInputStream();
                        OutputStream os = s.getOutputStream();
                        while (true) {
                                BufferedReader br = new BufferedReader(new
                                InputStreamReader(System.in));
                                String theString = br.readLine();
                                byte[] data = theString.getBytes();
                                String quit = new String("quit");
                                if (Arrays.equals(quit.getBytes(), data)) {
                                        System.out.println("Quit");
                                        break;
                                }
                                for (int i = 0; i < data.length; i++) {
                                        os.write(data[i]);
                                        int ch = is.read();
                                        System.out.print((char) ch);
                                }
                                System.out.println();
                        }
                        s.close();
                } catch (Exception e) {
                        System.out.print(e.toString());
                }
        }
}
```