Government College of Engineering & Textile Technology, Berhampore

Name : KRITIDIPTA GHOSH

Dept. : Computer Science & Engineering

Reg. No. : 026669 of 2019-2020

Roll No. : 11100119002

Semester: 6th

Year: 3rd year(2022)

Subject : Computer Networks Lab(PCC-CS692)

Programming Assignments

Kritidipta Ghosh, Government College of Engineering Textile Technology, Berhampore

04/06/2022

For all the source codes click the Clickable Link

Problem 1

Write a java program to read from a file.

Listing 1: Java Code

```java
// Implementation of Java File Reader.
// Developed on 02.03.2022
// Developed by Kritidipta Ghosh_11100119002
// Developed in Computer Networks Lab.

import java.io.*;

public class MyFileReader {
    public static void main(String[] args) throws IOException {
        FileReader fr = null;
        try {
            fr = new FileReader("input.txt");
        } catch (FileNotFoundException e) {
            System.out.println("File not found.");
        }
        int ch;
        while ((ch = fr.read()) != -1)
            System.out.print((char) ch);
        fr.close();
    }
}
```

Output:

Problem 2

Write a java program to write in a file.

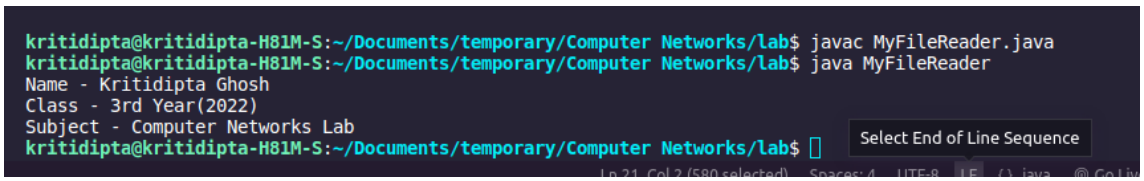

Figure 1: Output of the program File Reader

Listing 2: Java Code

```java
// Implementation of Java File Writer.
// Developed on 02.03.2022
// Developed by Kritidipta Ghosh_11100119002
// Developed in Computer Networks Lab.

import java.io.*;
import java.util.*;

public class MyFileWriter {
    public static void main(String[] args) throws IOException {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the text to be written:");
        String str = sc.nextLine();

        FileWriter fw = new FileWriter("output.txt");
        for (int i = 0; i < str.length(); i++) {
            fw.write(str.charAt(i));
        }
        System.out.println("Successfully written");
        fw.close();
        sc.close();
    }
}
```

Output:

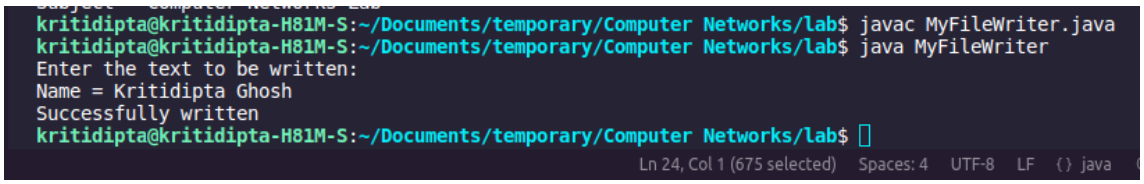

Figure 2: Output of the program File Writer

Problem 3

Implement Cyclic Redundancy Check with Java(Sender Side).

Listing 3: Java Code

```java
// Implementation of Java CRC(Sender side).
// Developed on 16.03.2022
// Developed by Kritidipta Ghosh_11100119002
// Developed in Computer Networks Lab.

import java.util.*;

class CRC {
    String sender(String data, String crc) {
        int n = data.length(), m = crc.length();
        char[] encoded = new char[n + m - 1];
        for (int i = 0; i < n; i++)
            encoded[i] = data.charAt(i);
        for (int i = n; i < n + m - 1; i++)
            encoded[i] = '0';

        for (int i = 0; i <= n;) {
            int j = 0;
            for (j = 0; j < m; j++) {
                encoded[i + j] = (encoded[i + j] == crc.charAt(j)) ? '0' :↩
                    '1';
            }
            for (; i <= n; i++) {
                if (encoded[i] == '1')
                    break;
            }
        }

        String ans = "";
        ans += data;
        for (int i = n; i < n + m - 1; i++)
            ans += encoded[i];
        System.out.println("Sender is sending the code:");
        System.out.println(ans);
        return ans;
    }
}

public class SenderMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the user data:");
        String user_data = sc.next();
        System.out.println("Enter the CRC generator:");
        String crc = sc.next();
        sc.reset();
```

```
46          CRC obj = new CRC();
47          String encoded = obj.sender(user_data, crc);
48          sc.close();
49      }
50  }
```

Output:

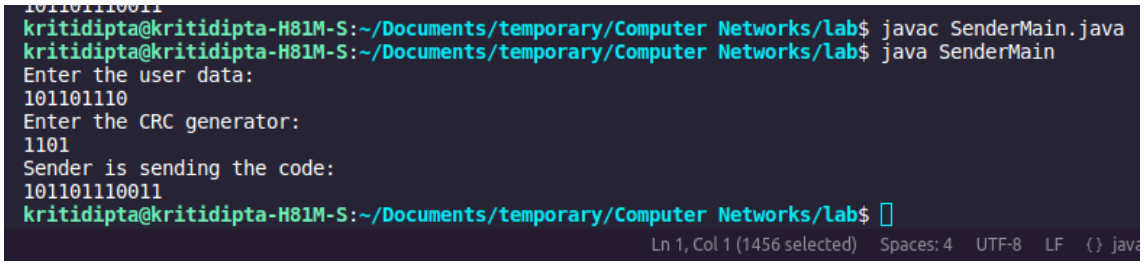

Figure 3: Output of the program CRC(Sender Side)

Problem 4

Implement Cyclic Redundancy Check with Java(Receiver Side).

Listing 4: Java Code

```
1  // Implementation of Java CRC(Receiver side).
2  // Developed on 16.03.2022
3  // Developed by Kritidipta Ghosh_11100119002
4  // Developed in Computer Networks Lab.
5
6  import java.util.*;
7
8  class CRC {
9      void Receiver(String codeword, String crc) {
10         int n = codeword.length(), m = crc.length();
11         char[] encoded = new char[n];
12         for (int i = 0; i < n; i++) {
13             encoded[i] = codeword.charAt(i);
14         }
15         n -= m;
16
17         for (int i = 0; i <= n;) {
18             int j = 0;
19             for (j = 0; j < m; j++) {
20                 encoded[i + j] = (encoded[i + j] == crc.charAt(j)) ? '0' :↩
                       '1';
```

```
            }
            for (; i <= n; i++) {
                if (encoded[i] == '1')
                    break;
            }
        }

        for (int i = n; i < n + m; i++) {
            if (encoded[i] == '1') {
                System.out.println("Error detected.");
                return;
            }
        }
        System.out.println("No error detected.");
    }
}

public class ReceiverMain {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the codeword:");
        String codeword = sc.next();
        System.out.println("Enter the crc generator:");
        String crc = sc.next();
        CRC obj = new CRC();
        obj.Receiver(codeword, crc);
        sc.close();
    }
}
```

Output:

```
kritidipta@kritidipta-H81M-S:~/Documents/temporary/Computer Networks/lab$ javac ReceiverMain.java
kritidipta@kritidipta-H81M-S:~/Documents/temporary/Computer Networks/lab$ java ReceiverMain
Enter the codeword:
101101110011
Enter the crc generator:
1101
No error detected.
kritidipta@kritidipta-H81M-S:~/Documents/temporary/Computer Networks/lab$ java ReceiverMain
Enter the codeword:
101111110011
Enter the crc generator:
1101
Error detected.
kritidipta@kritidipta-H81M-S:~/Documents/temporary/Computer Networks/lab$
                                                    Ln 50, Col 1 (1385 selected)  Spaces: 4  UTF-8  LF  {} java
```

Figure 4: Output of the program CRC (Receiver Side)

Problem 5

Implement Hamming Code with Java(Sender Side).

Listing 5: Java Code

```java
// Implementation of Hamming code(Sender Side).
// Developed on 23.03.2022
// Developed by Kritidipta Ghosh_11100119002
// Developed in Computer Networks Lab.

import java.util.*;

class HammingSender {
    int power(int base, int exp) {
        int ans = 1;
        while (exp > 0) {
            if (exp % 2 == 1) {
                exp--;
                ans *= base;
            } else {
                exp /= 2;
                base *= base;
            }
        }
        return ans;
    }

    String sender(String data) {
        int m = data.length();
        int r = 1;
        while (power(2, r) < m + r + 1)
            r++;

        // Setting the parity bits as -1:
        int[] codeword = new int[m + r + 1];
        for (int i = 0; i < r; i++) {
            if (power(2, i) < m + r + 1)
                codeword[power(2, i)] = -1;
        }

        // Setting the other bits.
        int idx = m - 1;
        for (int i = 1; i < m + r + 1; i++) {
            if (codeword[i] != -1) {
                codeword[i] = data.charAt(idx) - '0';
```

```
                idx--;
            }
        }

        // Now setting the parity bits accordingly:

        int parity = 0;
        while (parity < r) {
            int start = power(2, parity);
            parity++;
            int countOne = 0;

            for (int i = start; i < m + r + 1;) {
                if ((i / start) % 2 == 0)
                    i += start - 1;
                else if (codeword[i] == 1)
                    countOne++;
                i++;
            }
            codeword[start] = (countOne % 2 == 1) ? 1 : 0;
        }

        String ans = "";
        for (int i = m + r; i >= 1; i--)
            ans += (char) ('0' + codeword[i]);
        return ans;
    }
}

class HammingSenderMain {
    public static void main(String args[]) {
        String user_data;
        System.out.println("Enter the user data : ");
        Scanner sc = new Scanner(System.in);
        user_data = sc.next();

        HammingSender obj = new HammingSender();
        String codeWord = obj.sender(user_data);
        System.out.println("Sender is sending the code word : " + codeWord↩
            );
        sc.close();
    }
}
```

Output:

```
kritidipta@kritidipta-H81M-S:~/Documents/temporary/Computer Networks/lab$ javac HammingSenderMain.java
kritidipta@kritidipta-H81M-S:~/Documents/temporary/Computer Networks/lab$ java HammingSenderMain
Enter the user data :
1011001
Sender is sending the code word : 10101001110
kritidipta@kritidipta-H81M-S:~/Documents/temporary/Computer Networks/lab$
```

Figure 5: Output of the program Hamming Code(sender side)

Problem 6

Implement Hamming Code with Java(Receiver Side).

Listing 6: Java Code

```java
// Implementation of Hamming code(Receiver Side).
// Developed on 23.03.2022
// Developed by Kritidipta Ghosh_11100119002
// Developed in Computer Networks Lab.

import java.util.*;

class HammingReceiver {
    int power(int base, int exp) {
        int ans = 1;
        while (exp > 0) {
            if (exp % 2 == 1) {
                exp--;
                ans *= base;
            } else {
                exp /= 2;
                base *= base;
            }
        }
        return ans;
    }

    void print(int[] arr, int n) {
        for (int i = 1; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println("");
    }

    int receiver(String data) {
        int m = data.length();
        int[] codeword = new int[m + 1];
```

```
        int idx = m - 1;
        for (int i = 1; i < m + 1; i++) {
            codeword[i] = data.charAt(idx) - '0';
            idx--;
        }

        int r = 1;
        while (power(2, r) < m + 1)
            r++;
        m -= r;
        String ans = "";

        // chacking the parity bits:
        int parity = 0;
        while (parity < r) {
            int start = power(2, parity);
            parity++;
            int countOne = 0;

            for (int i = start; i < m + r + 1;) {
                if ((i / start) % 2 == 0)
                    i += start - 1;
                else if (codeword[i] == 1)
                    countOne++;
                i++;
            }
            ans += (countOne % 2 == 1) ? '1' : '0';
        }

        int bit = 0;
        for (int i = 0; i < ans.length(); i++) {
            if (ans.charAt(i) == '1')
                bit += power(2, i);
        }

        return bit;
    }
}

class HammingReceiverMain {
    public static void main(String args[]) {
        String codeword;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the codeword : ");
        codeword = sc.next();
        HammingReceiver obj = new HammingReceiver();
        int bit = obj.receiver(codeword);
```

```
80          if (bit != 0) {
81              System.out.println("Error occurred in the " + bit + "th bit of↩
                    the code word from right.");
82              System.out.print("The right codeword is : ");
83              int m = codeword.length();
84              for (int i = 0; i < m; i++) {
85                  if (i == m - bit)
86                      System.out.print((’1’ - codeword.charAt(i)));
87                  else
88                      System.out.print(codeword.charAt(i));
89              }
90              System.out.println("");
91          } else {
92              System.out.println("No error occurred.");
93          }
94          sc.close();
95      }
96  }
```

Output:

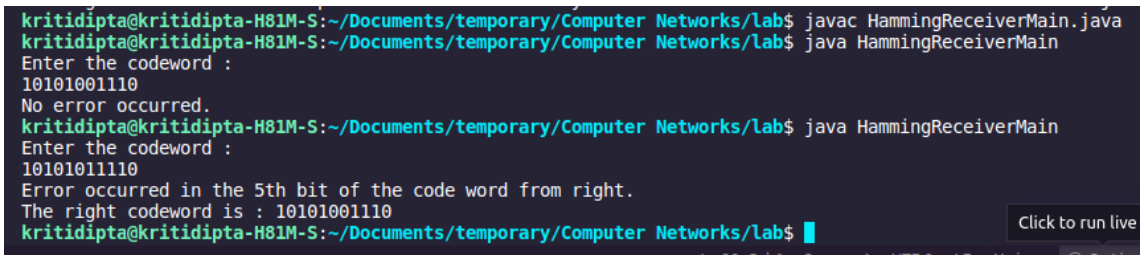

Figure 6: Output of the program Hamming code(receiver side)

Problem 7

Implement TCP Server with java.

Listing 7: Java Code

```
1  //Developed by - Kritidipta Ghosh
2  // Developed on - 13 <Apr> 2022
3
4  import java.io.*;
5  import java.net.*;
6
7  public class TCPServer {
8      private static Socket socket;
9
```

```
10      public static void main(String args[]) {
11          try {
12              System.out.println("Server started.");
13              int port = 2500;
14              ServerSocket serversocket = new ServerSocket(port);
15              socket = serversocket.accept();
16              System.out.println("Server started and listening to port 2500"↩
                    );
17              InputStream is = socket.getInputStream();
18              InputStreamReader isr = new InputStreamReader(is);
19              BufferedReader br = new BufferedReader(isr);
20              String number = br.readLine();
21              System.out.println("Received from client" + number + "\n");
22              BufferedReader bufferRead = new BufferedReader(new ↩
                    InputStreamReader(System.in));
23              String s = bufferRead.readLine();
24              OutputStream os = socket.getOutputStream();
25              OutputStreamWriter osw = new OutputStreamWriter(os);
26              BufferedWriter bw = new BufferedWriter(osw);
27              bw.write(s);
28              bw.flush();
29              System.out.println("Sent (to " + socket + ") client : " + s + ↩
                    "\n");
30          } catch (Exception e) {
31              e.printStackTrace();
32          } finally {
33              try {
34                  socket.close();
35              } catch (Exception e) {
36              }
37          }
38      }
39  }
```

Output:

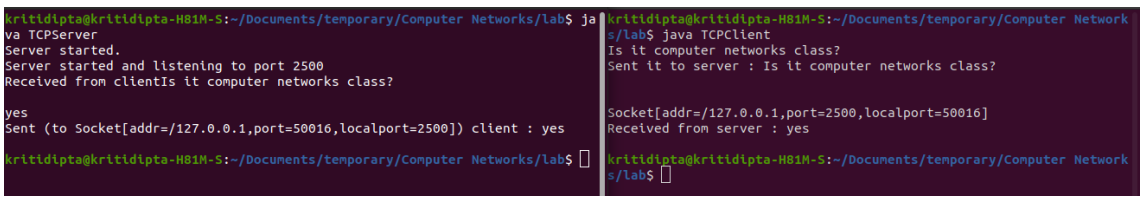

Figure 7: Output of the program TCP Client Server

Problem 8

Implement TCP client with Java.

Listing 8: Java Code

```java
//Developed by - Kritidipta Ghosh
// Developed on - 13 <Apr> 2022

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

public class TCPClient {
    private static Socket socket;

    public static void main(String args[]) {
        try {
            String host = "127.0.0.1";
            int port = 2500;
            socket = new Socket(host, port);

            OutputStream os = socket.getOutputStream();
            OutputStreamWriter osw = new OutputStreamWriter(os);
            BufferedWriter bw = new BufferedWriter(osw);

            BufferedReader bufferRead = new BufferedReader(new ↩
                InputStreamReader(System.in));
            String s = bufferRead.readLine();
            String sendMessage = s + "\n";
            bw.write(sendMessage);
            bw.flush();
            System.out.println("Sent it to server : " + sendMessage + "\n"↩
                );

            InputStream is = socket.getInputStream();
            InputStreamReader isr = new InputStreamReader(is);
            BufferedReader br = new BufferedReader(isr);
            System.out.println(socket);
            String message = br.readLine();
            System.out.println("Received from server : " + message + "\n")↩
                ;
        } catch (Exception exception) {
            exception.printStackTrace();
        }

        finally {
```

```
38              try {
39                  socket.close();
40              } catch (Exception e) {
41              }
42          }
43      }
44  }
```

Output:

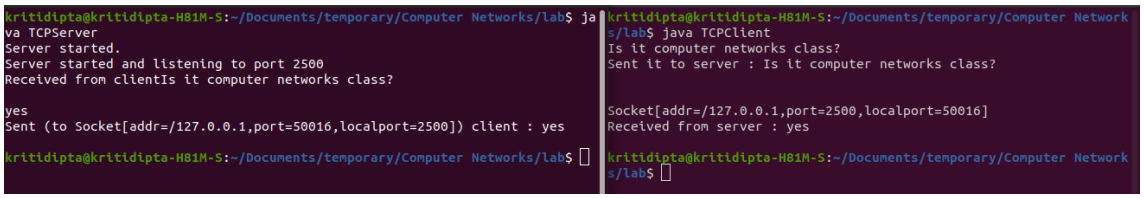

Figure 8: Output of the program TCP Client Server

Problem 9

Implement a simple TCP webserver with Java.

Listing 9: Java Code

```
1  // Developed by Kritidipta_Ghosh_11100119002
2  // Developed on - 18/05/2022
3
4  import java.util.*;
5  import java.io.*;
6  import java.net.*;
7
8  public class MyWebServer2{
9      private static ServerSocket serverSocket;
10
11     public static void main(String[] args) throws IOException {
12         serverSocket = new ServerSocket(8000);
13         while(true){
14             try{
15                 Socket s = serverSocket.accept();
16                 new ClientHandler(s);
17             } catch(Exception e){
18                 System.out.println(e);
19             }
20         }
21     }
```

```
22 }
23
24 class ClientHandler extends Thread{
25     private Socket socket;
26
27     public ClientHandler(Socket s){
28         socket = s;
29         start();
30     }
31
32     public void run(){
33         try{
34             BufferedReader in = new BufferedReader(new InputStreamReader(↩
                   socket.getInputStream()));
35             PrintStream out = new PrintStream(new BufferedOutputStream(↩
                   socket.getOutputStream()));
36
37             String s = in.readLine();
38             System.out.println(s);
39
40             String filename = "";
41             StringTokenizer st = new StringTokenizer(s);
42
43             try{
44                 if(st.hasMoreElements() && st.nextToken().equalsIgnoreCase↩
                       ("GET") && st.hasMoreElements()){
45                     filename = st.nextToken();
46                 } else{
47                     throw new FileNotFoundException();
48                 }
49
50                 if(filename.endsWith("/"))
51                     filename += "index.html";
52
53                 while(filename.indexOf("/") == 0)
54                     filename = filename.substring(1);
55
56                 filename = filename.replace('/', File.separator.charAt(0))↩
                       ;
57
58                 if(filename.indexOf("..")>=0 || filename.indexOf(":") >= 0↩
                        || filename.indexOf("|") >= 0)
59                     throw new FileNotFoundException();
60
61                 if(new File(filename).isDirectory()){
62                     filename = filename.replace('\\', '/');
63                     out.print("HTTP/1.0.301 Moved permanently\r\n"+"↩
```

```
                        Location: /"+filename+"/\r\n\r\n");
64                      out.close();
65                      return;
66                  }
67
68                  InputStream f = new FileInputStream(filename);
69                  String mimeType = "text/plain";
70                  if(filename.endsWith(".html") || filename.endsWith(".htm")↩
                        )
71                      mimeType = "text/html";
72                  else if(filename.endsWith(".jpg") || filename.endsWith(".↩
                        jpeg"))
73                      mimeType = "image/jpeg";
74                  else if(filename.endsWith(".gif"))
75                      mimeType = "image/gif";
76                  else if(filename.endsWith(".class"))
77                      mimeType = "application/octet-stream";
78                  out.print("HTTP/1.0 200 OK \r\n"+"Content type: "+mimeType↩
                        +"\r\n\r\n");
79
80                  byte[] a = new byte[4096];
81                  int n;
82                  while((n=f.read(a))>0)
83                      out.write(a,0,n);
84                  out.close();
85                  f.close();
86              } catch(FileNotFoundException x){
87                  out.println("HTTP/1.0 404 Not Found\r\n"+"Content type: ↩
                        text/html\r\n\r\n"+"<html><head></head><body>"+↩
                        filename+"not found</body></html>\n");
88                  out.close();
89              }
90
91
92          } catch(IOException x){
93              System.out.println(x);
94          }
95      }
96  }
```

Output:

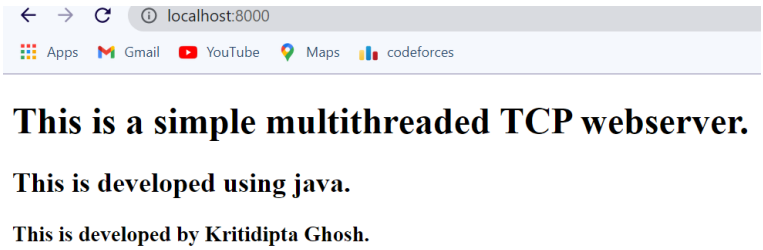

Figure 9: Output of the program TCP WebServer

Problem 10

Implement an UDP webserver with Java.

UDP Server:

Listing 10: Java Code

```
1  // Developed by Kritidipta_Ghosh_11100119002
2
3  import java.net.*;
4  class MyUDPServer
5  {
6      public static int serverPort = 998;
7      public static int clientPort = 999;
8      public static int buffer_size = 1024;
9      public static DatagramSocket ds;
10     public static byte buffer[] = new byte[buffer_size];
11
12     public static void TheServer() throws Exception
13     {
14         int pos = 0;
15         System.out.println("\n Enter text(q to exit)... \n");
16         while (true) {
17             int c = System.in.read();
18             switch (c) {
19                 case 'q':
20                     System.out.println("Server Quits.");
21                     return ;
22
23                 case '\r':
24                     break;
25
26                 case '\n':
27
28                     ds.send(new DatagramPacket(buffer, pos, InetAddress.↩
                           getLocalHost(), clientPort));
29
30
```

```
31                      ds.send(new DatagramPacket(buffer, pos, InetAddress.↩
                            getByName("172.168.1.22"), clientPort));
32                      ds.send(new DatagramPacket(buffer, pos, InetAddress.↩
                            getByName("172.168.1.23"), clientPort));
33
34
35                      ds.send(new DatagramPacket(buffer, pos, InetAddress.↩
                            getByName("255.255.255.255"), clientPort));
36
37                      pos = 0;
38                      break;
39
40                  default:
41                  buffer[pos++] = (byte) c;
42                      break;
43              }
44          }
45      }
46
47      public static void main(String[] args) throws Exception
48      {
49          ds = new DatagramSocket(serverPort);
50          TheServer();
51          ds.close();
52      }
53  }
```

UDP Client:

Listing 11: Java Code

```
1   // Developed by Kritidipta_Ghosh_11100119002
2
3   import java.net.*;
4   public class MyUDPClient {
5       public static int serverPort = 998;
6       public static int clientPort = 999;
7       public static int buffer_size = 1024;
8       public static DatagramSocket ds;
9       public static byte buffer[] = new byte[buffer_size];
10
11      public static void TheClient() throws Exception
12      {
13          while(true)
14          {
15              DatagramPacket p = new DatagramPacket(buffer, buffer.length);
16              ds.receive(p);
```

```
            System.out.println(new String(p.getData(), 0, p.getLength()));
        }
    }
    public static void main(String[] args) throws Exception
    {
        ds = new DatagramSocket(clientPort);
        TheClient();
    }
}
```