1. Write a client-server application that server would display a string sent from client

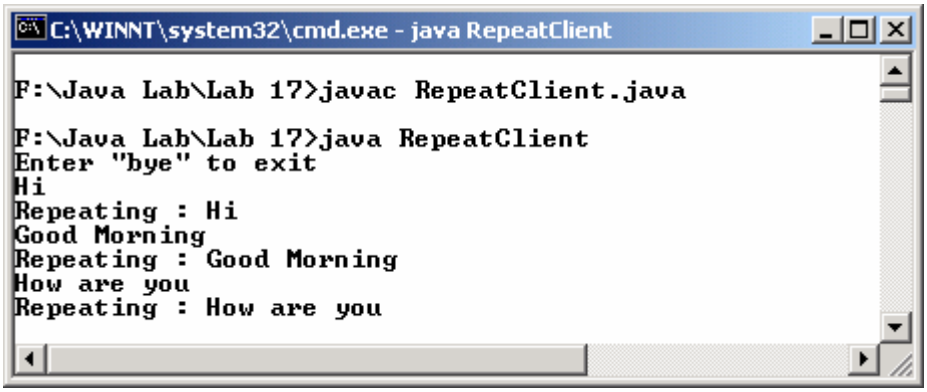
```
//Server
import java.io.*;
import java.net.*;

public class RepeatServer
{
   public static void main(String [] args)
   {
      try
      {
         System.out.println("Server started ..... listening on
port 8189");
         ServerSocket sckServer = new ServerSocket(8189);
         Socket sckClient = sckServer.accept();
         BufferedReader br = new BufferedReader(new
InputStreamReader(sckClient.getInputStream()));
         PrintWriter pw = new
PrintWriter(sckClient.getOutputStream(),true);

         String strLine;
         while(!(strLine = br.readLine()).equals("bye"))
         {
            pw.println("Repeating : " + strLine);
         }

         sckClient.close();
         sckServer.close();
      }
      catch(Exception e)
      {
         e.printStackTrace();
      }
   }
}
```

```
//Client
import java.net.*;
import java.io.*;

public class RepeatClient
{
   public static void main(String[] args)
   {
      Socket sckClient;
      String strHostname;

      BufferedReader brServer;
      BufferedReader brUser;
```

```
      PrintStream ps;
      String strLine;

      if(args.length > 0)
         strHostname = args[0];
      else
         strHostname = "localhost";

      try
      {
         sckClient = new Socket(strHostname, 8189);
         brServer = new BufferedReader(new
InputStreamReader(sckClient.getInputStream()));
         ps = new PrintStream(sckClient.getOutputStream());
         brUser = new BufferedReader(new
InputStreamReader(System.in));
         System.out.println("Enter \"bye\" to exit");

         while(true)
         {
            strLine = brUser.readLine();
            ps.println(strLine);

            if (strLine.equals("bye"))
               break;

               System.out.println(brServer.readLine());
         }
         sckClient.close();
         }
         catch (UnknownHostException e)
         {
            System.out.println(e);
         }
         catch (IOException e)
         {
            System.out.println(e);
         }
   }
}
```

The result at server:

The result at client:



2. Write a client/server application to send a file from client to server. Client accepts file path and IP of server from command line parameters.

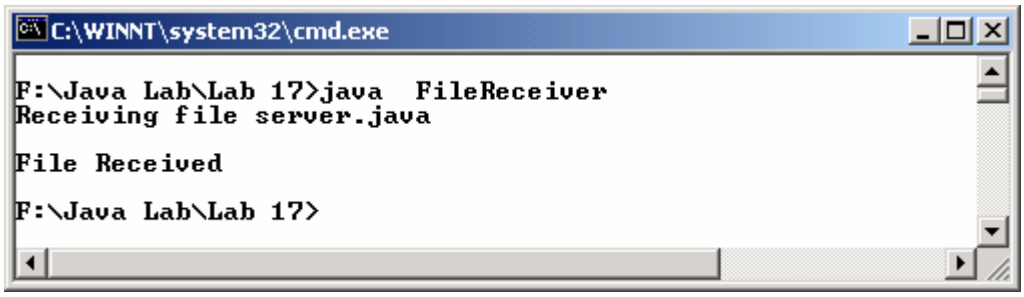
```
//Client
import java.net.*;
import java.io.*;

class FileSender
{
   public static void main(String[] args){

      if(args.length < 2)
      {
         System.out.println("Usage java FileSender <path of file>
<IP address>\njava FileSender \"F:\\Java Lab\\Lab
17\\FileSender.java\" 10.1.1.254");
         System.exit(0);
      }

      try
      {
         DatagramSocket sckSender = new DatagramSocket();
         InetAddress iaAddress = InetAddress.getByName(args[1]);
         int port = 4000;

         BufferedReader br = new BufferedReader(new
InputStreamReader(new FileInputStream(args[0])));
         String strLine = "";

         //retrieve the file name
         File filename = new File(args[0]);
         strLine = filename.getName();

         //send file name
```

```
        byte[] buf = strLine.getBytes();
        DatagramPacket fspacket = new
DatagramPacket(buf,buf.length,iaAddress, port);
        sckSender.send(fspacket);

        System.out.println("Transferring file " + args[0]);

        while((strLine = br.readLine()) != null)
        {
           buf = new byte[strLine.length()];
           buf = strLine.getBytes();
           fspacket = new
DatagramPacket(buf,buf.length,iaAddress, port);
           sckSender.send(fspacket);
        }

        System.out.println("File transfer complete");

        //send exit message to the server
        strLine = "exit";
        buf = strLine.getBytes();
        fspacket = new DatagramPacket(buf,buf.length,iaAddress,
port);
        sckSender.send(fspacket);

        sckSender.close();

      }catch(Exception e)
      {
         e.printStackTrace();
      }
   }
}
```

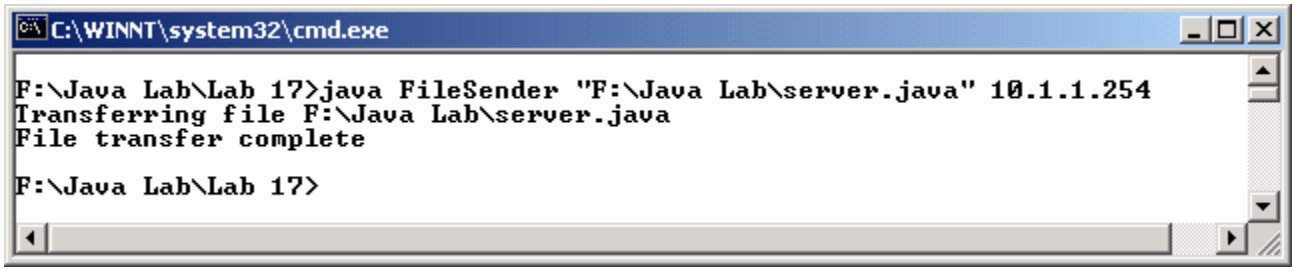
```
//Server

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

class FileReceiver
{
   public static void main(String[] args)
   {
      try
      {
         DatagramSocket sckReceiver = new DatagramSocket(4000);

         String strData = "";

         //get the filename
```

```
        byte[] buf = new byte[128];
        DatagramPacket pktReceiver = new DatagramPacket(buf,
buf.length);
        sckReceiver.receive(pktReceiver);
        strData = new String(pktReceiver.getData());

        PrintStream ps = new  PrintStream(new
FileOutputStream(strData));
        System.out.println("Receiving file " + strData);

        while(true)
        {
           buf = new byte[128];
           pktReceiver = new DatagramPacket(buf, buf.length);

           sckReceiver.receive(pktReceiver);
           strData = new String(pktReceiver.getData());

           //if client sends "exit" message, terminate the loop
           if(strData.trim().equals("exit"))
              break;

           ps.println(strData);
        }

        System.out.println("File Received");

        sckReceiver.close();

      }catch(Exception e)
      {
         e.printStackTrace();
      }
   }
}
```

The result at server



The result at client



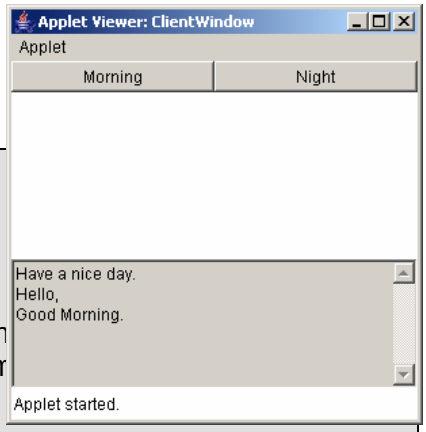
3.

```
import java.net.*;
import java.io.*;
public class OurSocket
{
   Socket socket;
   BufferedReader theReader;
   PrintWriter theWriter;
      public OurSocket(String host, int port)throws
UnknownHostException, IOException
   {
         this(new Socket(host, port));
   }
   public OurSocket(Socket s)throws UnknownHostException,
IOException
    {
         socket = s;
         theReader = new BufferedReader(
new InputStreamReader(socket.getInputStream()));
         theWriter = new PrintWriter(
socket.getOutputStream(), true);
   }
   public Socket getSocket()
   {
         return socket;
    }
   public void close()throws IOException
   {
         socket.close();
   }
    public String readLine()throws IOException
   {
         return theReader.readLine();
   }
      public void println(String s)throws IOException
   {
         theWriter.println(s);
   }
}
```

ServerWindow.java

```
import java.net.*;
import java.io.*;
import java.util.*;
public class ServerWindow
{
   public static final int PORT = 1234;
   ServerSocket sst;
   OurSocket os;
   Vector morning = new Vector();
   Vector night   = new Vector();
   public static void main(String [] args)
   {
         new ServerWindow().init();
   }
   public void init()
   {
         try
      {
            BufferedReader in;
               String line;
                in = new BufferedReader(new InputStreamReader(new
FileInputStream("morning.txt")));
            while ((line = in.readLine()) != null)
         {
               morning.addElement(line);
            }
   in.close();

         in = new BufferedReader(new InputStreamReader(new
FileInputStream("night.txt")));
   while ((line = in.readLine()) != null)
   {
            night.addElement(line);
      }

   in.close();

         sst = new ServerSocket(PORT);
      while(true)
   {
         System.err.println("Here's the Server ... ready and
running.");
               os = new OurSocket(sst.accept());
         String s = os.readLine();
         System.out.println("Read " + s + " from client.");
   if (s.equalsIgnoreCase("morning"))
   {
      send(morning);
      }
      else if (s.equalsIgnoreCase("night"))
```

```
   {
         send(night);
      }
      else
   {
         System.err.println("Invalid request: " + s);
      }
            os.close();
      System.err.println("Finished processing !!");
  }
  }
  catch(Exception ex)
{
   ex.printStackTrace();
}
}
public void send(Vector v) throws IOException
{
   int size = v.size();
   for(int i = 0; i < 3; i++)
   {
         int n = (int) (Math.random() * size);
         os.println((String) v.elementAt(n));
   }
  }
}
```



ClientWindow.java

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.net.*;
import java.io.*;
//<applet code=ClientWindow width=300 heigh
public class ClientWindow extends Applet im
ActionListener
{
   Button morningButton = new Button("Morning");
   Button nightButton = new Button("Night");
   TextArea message = new TextArea(5,20);
   OurSocket os;
   public void init()
   {
         setLayout(new BorderLayout());
         Panel p = new Panel();
         add(p, BorderLayout.NORTH);
         add(message, BorderLayout.SOUTH);
```

```
        p.setLayout(new GridLayout(1, 0));
        p.add(morningButton);
        p.add(nightButton);
        message.setEditable(false);
        morningButton.addActionListener(this);
        nightButton.addActionListener(this);
   }
   public void fatalError(Exception ex)
   {
        ex.printStackTrace();
        try
     {
             os.close();
        }
        catch (Exception ex1)
     {
            ;
        }
   }
   public void actionPerformed(ActionEvent e)
   {
        if(e.getSource() == morningButton)
     {
             try
        {
                message.setText("");
                os = new OurSocket(getCodeBase().getHost(),
1234);
                os.println("Morning");
                message.append(os.readLine() + "\n");
                message.append(os.readLine() + "\n");
                message.append(os.readLine() + "\n");
                os.close();
             }
     catch (Exception ex)
     {
             fatalError(ex);
          }
     }
     else
   {
          try
     {
             message.setText("");
             os = new OurSocket(getCodeBase().getHost(), 1234);
             os.println("Night");
             message.append(os.readLine() + "\n");
             message.append(os.readLine() + "\n");
             message.append(os.readLine() + "\n");
             os.close();
          }
```

```
            catch (Exception ex)
      {
               fatalError(ex);
            }
      }
   }
}
```

Do It Yourself

4.1. Do workshop 7

4.2. Write a program to download a file from an specified URL (using console or swing as you like)

4.3. Write a simple chat application to message between two computers with swing interface.

Self-study Samples

+ Sample supplied programs:
  - URL
  - MessageClientServer
  - WebServer
  - ZipClientServer

+ Sample application "Download Manager" on java2s.com
  http://www.java2s.com/Code/Java/Tiny-Application/Download-Manager.htm

+ java2s.com

+ Java tutorials

+ javapassion.com