```
TCP/IP sockets, Client - Server program to make the client send the
file name and to make the server send back the contents of the
requested file if present.
```

import java.net.*;

import java.io.*;

public class ContentsClient

{

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

  {

    Socket sock = new Socket("127.0.0.1", 4000);

    // reading the file name from keyboard. Uses input stream

    System.out.print("Enter the file name");

    BufferedReader keyRead = new BufferedReader(new   InputStreamReader(System.in));

    String fname = keyRead.readLine();

    // sending the file name to server. Uses PrintWriter

    OutputStream ostream = sock.getOutputStream( );

    PrintWriter pwrite = new PrintWriter(ostream, true);

    pwrite.println(fname);

    // receiving the contents from server. Uses input stream

    InputStream istream = sock.getInputStream();

    BufferedReader socketRead = new BufferedReader(new InputStreamReader(istream));

    String str;

    while((str = socketRead.readLine()) != null) // reading line-by-line

    {

      System.out.println(str);

    }

    pwrite.close();

    socketRead.close();

    keyRead.close();

  }

}

TCP/IP sockets, Client – Server program to make the client send the file name and to make the server send back the contents of the requested file if present.

```java
import java.net.*;

import java.io.*;

public class ContentsServer
{
  public static void main(String args[]) throws Exception
  {
    // establishing the connection with the server
    ServerSocket sersock = new ServerSocket(4000);

    System.out.println("Server ready for connection");

    Socket sock = sersock.accept();    // binding with port: 4000

    System.out.println("Connection is successful and wating for client request");


    // reading the file name from client
    InputStream istream = sock.getInputStream( );

    BufferedReader fileRead =new BufferedReader(new InputStreamReader(istream));

    String fname = fileRead.readLine( );


    // reading file contents
    BufferedReader contentRead = new BufferedReader(new FileReader(fname) );


    // keeping output stream ready to send the contents
    OutputStream ostream = sock.getOutputStream( );

    PrintWriter pwrite = new PrintWriter(ostream, true);


    String str;

    while((str = contentRead.readLine()) !=  null) // reading line-by-line from file
    {
        pwrite.println(str);  // sending each line to client
    }
     sock.close();  sersock.close(); // closing network sockets

    pwrite.close();  fileRead.close(); contentRead.close();
  }
}
```