

```
1 import java.io.*;
2 import java.net.Socket;
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class client {
7     public static void main(String[] args) {
8         try {
9             System.out.println("Connecting to the
server...");
10
11             // Create a socket and connect to the
server on localhost, port 8080
12             Socket socket = new Socket("localhost",
8080);
13
14             System.out.println("Connected to the
server.");
15
16             // Create input and output streams for
communication with the server
17             BufferedReader in = new BufferedReader(
new InputStreamReader(socket.getInputStream()));
18             PrintWriter out = new PrintWriter(socket.
getOutputStream(), true);
19
20             // Specify the folder path on the client
side
21             String desktopPath = System.getProperty("
user.home") + "/Desktop" + "/Files";
22             File folder = new File(desktopPath);
23
24             // List files in the folder
25             File[] files = folder.listFiles();
26             if (files == null) {
27                 System.err.println("No files found in
the folder: " + desktopPath);
28                 return;
29             }
30
31             // Send up to 20 files at a time
```

```
32         int filesToSend = Math.min(20, files.  
length);  
33         for (int i = 0; i < filesToSend; i++) {  
34             File file = files[i];  
35  
36             // Send the file name to the server  
37             out.println(file.getName());  
38  
39             // Read the file content and send it  
to the server  
40             try (BufferedReader fileReader = new  
BufferedReader(new FileReader(file))) {  
41                 String line;  
42                 while ((line = fileReader.  
readLine()) != null) {  
43                     out.println(line);  
44                 }  
45             }  
46  
47             System.out.println("File sent: " +  
file.getName());  
48         }  
49  
50         // Close the streams and socket  
51         in.close();  
52         out.close();  
53         socket.close();  
54  
55         } catch (IOException e) {  
56             System.err.println("Connection failed.  
Make sure the server is running and check your  
firewall settings.");  
57             e.printStackTrace();  
58         }  
59     }  
60 }  
61
```