

```

1 // fo1/9/copy/Copy1.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 import java.io.*;
7
8 /**
9  * Simple read-a-char, write-a-char loop to exercise file I/O.
10  *
11  * Usage: java Copy1 inputFile outputFile
12  */
13
14 public class Copy1
15 {
16     private static final int EOF = -1; // end of file character rep.
17
18     /**
19      * All work is done here.
20      *
21      * @param args names of the input file and output file.
22      */
23
24     public static void main( String[] args )
25     {
26         FileReader inStream = null;
27         FileWriter outStream = null;
28         int ch;
29
30         try {
31             // open the files
32             inStream = new FileReader( args[0] );
33             outStream = new FileWriter( args[1] );
34
35             // copy
36             while ((ch = inStream.read()) != EOF) {
37                 outStream.write( ch );
38             }
39         }
40         catch (IndexOutOfBoundsException e) {
41             System.err.println(
42                 "usage: java Copy1 sourcefile targetfile" );
43         }
44         catch (FileNotFoundException e) {
45             System.err.println( e ); // rely on e's toString()
46         }
47         catch (IOException e) {
48             System.err.println( e );
49         }
50         finally { // close the files
51             try {
52                 if (inStream != null) {
53                     inStream.close();
54                 }
55             }
56             catch (Exception e) {

```

```

57         }
58     }
59     try {
60         if (outStream != null) {
61             outStream.close();
62         }
63     }
64     catch (Exception e) {
65         System.err.println("Unable to close output stream.");
66     }
67 }
68 }
69 }

```

```

1 // fo1/9/copy/Copy2.java
2 //
3 //
4 // Copyright 2003 Bill Campbell and Ethan Bolker
5
6 import java.io.*;
7
8 /**
9  * Simple read-a-line write-a-line loop to exercise file I/O.
10  *
11  * Usage: java Copy2 inputFile outputFile
12  */
13
14 public class Copy2
15 {
16     /**
17      * All work is done here.
18      *
19      * @param args names of the input file and output file.
20      */
21
22     public static void main( String[] args )
23     {
24         BufferedReader inStream = null;
25         BufferedWriter outStream = null;
26         String line;
27
28         try {
29             // open the files
30             inStream = new BufferedReader(new FileReader(args[0]));
31             outStream = new BufferedWriter(new FileWriter(args[1]));
32
33             // copy
34             while ((line = inStream.readLine()) != null) {
35                 outStream.write( line );
36                 outStream.newLine();
37             }
38         }
39         catch (IndexOutOfBoundsException e) {
40             System.err.println(
41                 "usage: java Copy2 sourcefile targetfile" );
42         }
43         catch (FileNotFoundException e) {
44             System.err.println( e ); // rely on e's toString()
45         }
46         catch (IOException e) {
47             System.err.println( e );
48         }
49         finally { // close the files
50             try {
51                 if (inStream != null) {
52                     inStream.close();
53                 }
54             }
55             catch (Exception e) {
56                 System.err.println("Unable to close input stream.");

```

```

57     }
58     try {
59         if (outStream != null) {
60             outStream.close();
61         }
62     }
63     catch (Exception e) {
64         System.err.println("Unable to close output stream.");
65     }
66 }
67 }
68 }

```