

```

package ir.ac.mahshahriau.javaclass.test;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

// Using Byte Streams
public class IO1 {
    public static void main(String[] args) throws IOException {

        FileInputStream in = null;
        FileOutputStream out = null;
        int count = 0;

        try {
            in = new FileInputStream("c:\\in.txt");
            out = new FileOutputStream("c:\\out.txt");
            int c;

            while ((c = in.read()) != -1) { // Reads one byte
                count++;
                out.write(c);
            }
        }
        catch(Exception e)
        {

        }
        finally { // Always Close Streams
            if (in != null) {
                in.close();
            }
            if (out != null) {
                out.close();
            }
        }
        System.out.println(count);
    }
} // Byte streams should only be used for the most primitive I/O

```

```

package ir.ac.mahshahriau.javaclass.test;

import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

// Using Character Streams
public class IO2 {
    public static void main(String[] args) throws IOException {

        FileReader in = null;
        FileWriter out = null;
        int count = 0;

        try {
            in = new FileReader("c:\\in.txt");
            out = new FileWriter("c:\\out.txt");

```

```

        int c;
        while ((c = in.read()) != -1) {
            count++;
            out.write(c);
        }
    } finally {
        if (in != null) {
            in.close();
        }
        if (out != null) {
            out.close();
        }
    }
    System.out.println(count);
}
}

```

```
package ir.ac.mahshahriau.javaclass.test;
```

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

```

```
// Buffered Streams, more efficient
```

```
public class IO3 {
```

```
    public static void main(String[] args) throws IOException {
```

```

        BufferedReader in = null;
        BufferedWriter out = null;
    
```

```
    try {
```

```

        in = new BufferedReader(new FileReader("c:\\in.txt"));
        out = new BufferedWriter(new FileWriter("d:\\out.txt"));
    
```

```

        int c;
        while ((c = in.read()) != -1) {
            out.write(c);
        }
    
```

```
    } finally {
```

```

        if (in != null) {
            in.close();
        }
    
```

```

        if (out != null) {
            out.close();
        }
    
```

```
    }
```

```
}
```

```
package ir.ac.mahshahriau.javaclass.test;
```

```

import java.io.FileReader;
import java.io.FileWriter;

```

```

import java.io.BufferedReader;
import java.io.PrintWriter;
import java.io.IOException;

// Line-Oriented I/O
public class IO4 {
    public static void main(String[] args) throws IOException {

        BufferedReader in = null;
        PrintWriter out = null;
        int count = 0;

        try {
            in = new BufferedReader(new FileReader("c:\\in.txt"));
            out = new PrintWriter(new FileWriter("c:\\out.txt"));

            String l;
            while ((l = in.readLine()) != null) { // Reads one line
                count++;
                out.println(l + " " + l);
            }
        } finally {
            if (in != null) {
                in.close();
            }
            if (out != null) {
                out.close();
            }
        }
        System.out.println(count);
    }
}

package ir.ac.mahshahriau.javaclass.test;

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

// Breaking Input into Tokens
public class IO5 {
    public static void main(String[] args) {
        Scanner s = null;
        try {
            s = new Scanner(new BufferedReader(new FileReader("c:\\in.txt")));
            s = new Scanner("This is for test");

            while (s.hasNext()) {
                System.out.println(s.next());
            }
        } catch (FileNotFoundException e) { // Using catch instead of throws
            e.printStackTrace();
        } finally {
            if (s != null) {
                s.close();
            }
        }
    }
}

```