File Output

zyBook Chap 6.5

Recap – How do we read a file?

- **Step 1:** Specify the **file path** as a **String** object

```
String fileName = "data.txt";
```

- **Step 2:** Construct a **File** object to get the information about a file on the disk

```
import java.io.File;
File inputFile = new File(fileName);
```

- **Step 3:** Construct a **Scanner** object to read the file

```
import java.util.Scanner;
Scanner scnr = new Scanner(inputFile);
```

How do we write to a file?

- **Step 1:** Specify the **file path** as a **String** object

```
String fileName = "output.txt";
```

- **Step 2:** Construct a **File** object to get the information about a file on the disk

```
import java.io.File;
File outputFile = new File(fileName);
```

- **Step 3:** Construct a **PrintStream** object to print to the file

```
import java.io.PrintStream;
PrintStream out = new PrintStream(outputFile);
```

```
PrintStream <name> = new PrintStream(new File("<fileName>"));
```

PrintStream Objects

- The console output object, **System.out**, is a **PrintStream** object

```
// Prints the message to the console, equivalent to System.out.println()
PrintStream consoleOutput = System.out;
consoleOutput.println("Message in console!");

// Prints the message to the file result.txt
PrintStream fileOutput = new PrintStream(new File("result.txt"));
fileOutput.println("Message in file!");
```

- Any methods we have used with **System.out** (e.g., **print**, **println**, **printf**) also work with **PrintStream** objects

Write to a File

```
PrintStream <name> = new PrintStream(new File("<fileName>"));
```

- If the given file **does not exist**, it is **created**

```java
import java.io.File;
import java.io.PrintStream;

public class PrintToFile {
    public static void main (String[] args) {
        // Specify the file that we would like to print to
        File outFile = new File("/Users/ginabai/Desktop/HelloWorld.txt");

        // Construct the PrintStream object to print to the specified output file
        PrintStream output = new PrintStream(outFile);
        output.println("Hello!");
        output.printf("Prints the number %d ", 2023);
        output.print("to the output file!");
        output.close();
    }
}
```

```
$ javac PrintToFile.java
PrintToFile.java:10: error: unreported exception FileNotFoundException;
must be caught or declared to be thrown
        PrintStream output = new PrintStream(outFile);
                             ^
1 error
```

```java
import java.io.File;
import java.io.PrintStream;
import java.io.FileNotFoundException;

public class PrintToFile {
    public static void main (String[] args) throws FileNotFoundException {
        // Specify the file that we would like to print to
        File outFile = new File("/Users/ginabai/Desktop/HelloWorld.txt");

        // Construct the PrintStream object to print to the specified output file
        PrintStream output = new PrintStream(outFile);
        output.println("Hello!");
        output.printf("Prints the number %d ", 2023);
        output.print("to the output file!");
        output.close();
    }
}
```

```
Desktop % javac PrintToFile.java
Desktop % java PrintToFile
Desktop %
```

Write to a File

```
PrintStream <name> = new PrintStream(new File("<fileName>"));
```

- If the given file **does not exist**, it is **created**
- If the given file **already exists**, it is **overwritten**

```java
import java.io.File;
import java.io.PrintStream;
import java.io.FileNotFoundException;

public class PrintToFile {
    public static void main (String[] args) throws FileNotFoundException {
        // Specify the file that we would like to print to
        File outFile = new File("/Users/ginabai/Desktop/HelloWorld.txt");

        // Construct the PrintStream object to print to the specified output file
        PrintStream output = new PrintStream(outFile);
        output.println("Hello!");
        output.printf("Prints the number %.1f ", 2023.327);
        output.print("to the output file!");
        output.close();
    }
}
```

```
Desktop % javac PrintToFile.java
Desktop % java PrintToFile
Desktop %
```

HelloWorld.txt

```
Hello!
Prints the number 2023.3 to the output file!
```

Write to a File

```
PrintStream <name> = new PrintStream(new File("<fileName>"));
```

- If the given file **does not exist**, it is **created**
- If the given file **already exists**, it is **overwritten**
  - How to avoid overwriting a file?
    - <outputFile>.exists()
      - Common options for an output file that already exists: overwrite the file, stop the program, prompt for a new output file name, prompt user whether they would like to overwrite

```java
import java.io.File;
import java.io.PrintStream;
import java.io.FileNotFoundException;

public class PrintToFile {
    public static void main (String[] args) throws FileNotFoundException {
        // Specify the file that we would like to print to
        File outFile = new File("/Users/ginabai/Desktop/HelloWorld.txt");

        if(outFile.exists()){
            System.out.println("Output file exists. It was not overwritten.");
            System.exit(1); // Terminate the program execution
        }

        // Construct the PrintStream object to print to the specified output file
        PrintStream output = new PrintStream(outFile);
        output.println("Hello!");
        output.printf("Prints the number %.1f ", 2023.327);
        output.print("to the output file!");
        output.close();
    }
}
```

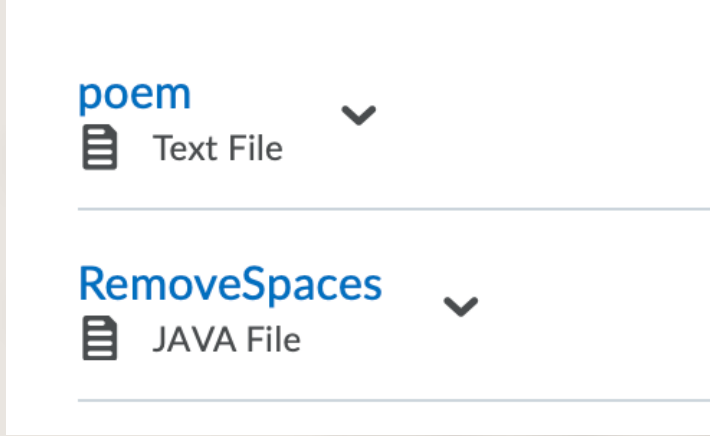
```
$ javac PrintToFile.java
$ java PrintToFile
Output file exists. It was not overwritten.
```

Write to a File

```
PrintStream <name> = new PrintStream(new File("<fileName>"));
```

- If the given file **does not exist**, it is **created**
- If the given file **already exists**, it is **overwritten**
- **Do not open a file** for **reading** (i.e., Scanner) and **writing** (i.e., PrintStream) **at the same time**
  - You could overwrite your input file by accident! The result can be an empty file (size 0 bytes).

Coding Practice – File I/O



- Write a program called **RemoveSpaces** that
  - Prompt the user for an input file name
  - If the input file does not exist, re-prompt the user for an input file name
  - Otherwise,
    - If the output file does not exist, it **copies each line of the input file to the output file with no whitespace before or after each token**.
    - If the output file does exist, it outputs "<fileName> already exists!" and **does not overwrite** the file.

poem.txt

Still I Rise

Maya Angelou

You may write me down in history
With your bitter, twisted lies,
You may trod me in the very dirt
But still, like dust, I'll rise.

poemNoSpace.txt

StillIRise

MayaAngelou

Youmaywritemedowninhistory
Withyourbitter,twistedlies,
Youmaytrodmeintheverydirt
Butstill,likedust,I'llrise.

Sample Solution

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

public class RemoveSpaces {
    public static void main(String[] args) throws FileNotFoundException {
        Scanner console = new Scanner(System.in);

        System.out.print("Enter file name: ");
        String fileName = console.nextLine().trim();

        File inputFile = new File(fileName);

        // Use a while loop to validte the input file name
        while(!inputFile.exists()){
            System.out.print("Input file does not exist, try again: ");
            fileName = console.nextLine().trim();
            inputFile = new File(fileName);
        }
        // Construct a Scanner to read the input file
        Scanner input = new Scanner(inputFile);

        // Specify the output file name
        File outputFile = new File("poemNoSpace.txt");

        // Use an if statement to check whether the output file already exists
        if (outputFile.exists()) {
            System.out.println("poemNoSpace.txt already exists!");
            System.exit(1);
        }
        // Construct a PrintStream to print to the output file
        PrintStream output = new PrintStream(outputFile);

        // Line-based processing
        while (input.hasNextLine()) {
            String line = input.nextLine();
            Scanner lineScnr = new Scanner(line);
            while (lineScnr.hasNext()) {
                // remove the spaces by reading and printing the tokens
                output.print(lineScnr.next());
            }
            output.println();
            lineScnr.close(); // Close the Scanner for the line
        }

        input.close();  // Close the Scanner for the input file
        output.close(); // Close the PrintStream for the output file
    }
}
```