Chapter Summary

Files are represented in Java as `File` objects. The `File` class is found in the `java.io` package.

A `Scanner` object can read input from a file rather than from the keyboard. This task is achieved by passing `new File(`*filename*`)` to the `Scanner`'s constructor, rather than passing `System.in`.

A checked exception is a program error condition that must be caught or declared in order for the program to compile. For example, when constructing a `Scanner` that reads a file, you must write the phrase `throws FileNotFoundException` in the `main` method's header.

The `Scanner` treats an input file as a one-dimensional stream of data that is read in order from start to end. The input cursor consumes (moves past) input tokens as they are read and returns them to your program.

A `Scanner` that reads a file makes use of the various `hasNext` methods to discover when the file's input has been exhausted.

`Scanner`s can be passed as parameters to methods to read part or all of a file, since they are objects and therefore use reference semantics.

A file name can be specified as a relative path such as `data/text/numbers.dat`, which refers to a file called `numbers.dat` that exists in the `data/text/` subfolder of the current directory. Alternatively, you can specify a full file path such as `C:/Documents and Settings/user/My Documents/data/text/numbers.dat`.

In many files, input is structured by lines, and it makes sense to process those files line by line. In such cases, it is common to use nested loops: an outer loop that iterates over each line of the file and an inner loop that processes the tokens in each line.

Output to a file can be achieved with a `PrintStream` object, which is constructed with a `File` and has the same methods as `System.out`, such as `println` and `print`.

Self-Check Problems

Section 6.1: File-Reading Basics

1. What is a file? How can we read data from a file in Java?
   Rather? You can create a File object and represent a file
2. What is wrong with the following line of code?

```
Scanner input = new Scanner("test.dat");
```

   It will scan the string "test.dat", not the file of that name
3. Which of the following is the correct syntax to declare a `Scanner` to read the file `example.txt` in the current directory?

   a. `Scanner input = new Scanner("C:\example.txt");`
   b. `Scanner input = new Scanner(new File("example.txt"));`
   c. `Scanner input = new File("\\example.txt");`
   d. `File input = new Scanner("/example.txt");`
   e. `Scanner input = new Scanner("C:/example.txt");`

4. Write code to construct a `Scanner` object to read the file `input.txt`, which exists in the same folder as your program.

Scanner input = new Scanner(new File("input.txt"));

Section 6.2: Details of Token-Based Processing

5. Given the following line of input, what tokens does a `Scanner` break the line apart into?

```
welcome...to the matrix.
```

a. `"welcome", "to", "the", "matrix"`
b. `"welcome...to the matrix."`
c. `"welcome...to", "the", "matrix."`
d. `"welcome...", "to", "the matrix."`
e. `"welcome", "to the matrix"`

6. Given the following line of input, what tokens does a `Scanner` break the line apart into?

```
in fourteen-hundred 92
columbus sailed the ocean blue :)
```

a. `"in", "fourteen-hundred", "92"`
b. `"in", "fourteen-hundred", "92", "columbus", "sailed", "the", "ocean", "blue", ":)"`
c. `"in", "fourteen", "hundred", "92", "columbus", "sailed", "the", "ocean", "blue"`
d. `"in", "fourteen-hundred", "92\ncolumbus", "sailed", "the", "ocean", "blue :)"`
e. `"in fourteen-hundred 92", "columbus sailed the ocean blue :)"`

7. How many tokens are there in the following input, and what `Scanner` method(s) can be used to read each of the tokens?

17 tokens. Scanner.next() can read any token, or you could also use the hasTypeOne() and get the next token using hasTypeOne() and get that

```
Hello there,how are you?
I am "very well", thank you.
12 34 5.67 (8 + 9) "10"
```

8. What is wrong with the following line of code?

```
Scanner input = new Scanner(new File("C:\temp\new files\test.dat"));
```

(Hint: Try printing the `String` in this line of code.) The backslashes are not escaped

9. Answer the following questions about a Java program located on a Windows machine in the folder `C:\Documents and Settings\amanda\My Documents\programs`:

What? I literally don't understand what this question is asking

a. What are two legal ways you can refer to the file `C:\Documents and Settings\amanda\My Documents\programs\numbers.dat`?
b. How can you refer to the file `C:\Documents and Settings\amanda\My Documents\programs\data\homework6\input.dat`?
c. How many, and in what legal, ways can you refer to the file `C:\Documents and Settings\amanda\My Documents\homework\data.txt`?

10. Answer the following questions about a Java program located on a Linux machine in the folder `/home/amanda/Documents/hw6`:

a. What are two legal ways you can refer to the file `/home/amanda/Documents/hw6/names.txt`? names.txt or the full file path
b. How can you refer to the file `/home/amanda/Documents/hw6/data/numbers.txt`? data/numbers.txt
c. How many legal ways can you refer to the file `/home/amanda/download/saved.html`? ∞ - while it would be impractical you could refer to it an infinite # of ways eg /home/amanda/../amanda/../

11. The following program contains 6 mistakes! What are they?

```
1  public class Oops6 {
2      public static void main(String[] args) {
3          Scanner in = new Scanner("example.txt");
4          countWords(in);
5      }
6
7      // Counts total lines and words in the input scanner.
8      public static void countWords(Scanner input) {
9          Scanner input = new Scanner("example.txt");
10         int lineCount = 0;
11         int wordCount = 0;
12
13         while (input.nextLine()) {
14             String line = input.line();    // read one line
15             lineCount++;
16             while (line.next()) {          // tokens in line
17                 String word = line.hasNext;
18                 wordCount++;
19             }
20         }
21     }
22 }
```

Section 6.3: Line-Based Processing

12. For the next several questions, consider a file called `readme.txt` that has the following contents:

```
6.7        This file has
        several input lines.

  10 20         30   40

test
```

What would be the output from the following code when it is run on the `readme.txt` file?

```
Scanner input = new Scanner(new File("readme.txt"));
int count = 0;
while (input.hasNextLine()) {
    System.out.println("input: " + input.nextLine());
    count++;
}
System.out.println(count + " total");
```

13. What would be the output from the code in the previous exercise if the calls to `hasNextLine` and `nextLine` were replaced by calls to `hasNext` and `next`, respectively?

14. What would be the output from the code in the previous exercise if the calls to `hasNextLine` and `nextLine` were replaced by calls to `hasNextInt` and `nextInt`, respectively? How about `hasNextDouble` and `nextDouble`?

15. Write a program that prints itself to the console as output. For example, if the program is stored in `Example.java`, it will open the file `Example.java` and print its contents to the console.

16. Write code that prompts the user for a file name and prints the contents of that file to the console as output. Assume that the file exists. You may wish to place this code into a method called `printEntireFile`.

17. Write a program that takes as input lines of text like the following:

```
This is some
text here.
```

The program should produce as output the same text inside a box, as in the following:

```
+--------------+
| This is some |
| text here.   |
+--------------+
```

Your program will have to assume some maximum line length (e.g., 12 in this case).

Section 6.4: Advanced File Processing

18. What object is used to write output to a file? What methods does this object have available for you to use?

19. Write code to print the following four lines of text into a file named `message.txt`:

```
Testing,
1, 2, 3.

This is my output file.
```

20. Write code that repeatedly prompts the user for a file name until the user types the name of a file that exists on the system. You may wish to place this code into a method called `getFileName`, which will return that file name as a `String`.

21. In Problem 16, you wrote a piece of code that prompted the user for a file name and printed that file's contents to the console. Modify your code so that it will repeatedly prompt the user for the file name until the user types the name of a file that exists on the system.

Exercises

1. Write a method called `boyGirl` that accepts a `Scanner` that is reading its input from a file containing a series of names followed by integers. The names alternate between boys' names and girls' names. Your method should compute the absolute difference between the sum of the boys' integers and the sum of the girls' integers. The input could end with either a boy or girl; you may not assume that it contains an even number of names. For example, if the input file contains the following text:

```
Erik 3 Rita 7 Tanner 14 Jillyn 13 Curtis 4 Stefanie 12 Ben 6
```