

Chapter 5:

Loops and Files

**Starting Out with Java:
Early Objects
Fourth Edition**

by Tony Gaddis

Reading Data From a File

- You use the `File` class and the `Scanner` class to read data from a file:

Pass the name of the file as an argument to the `File` class constructor.

```
File myFile = new File("Customers.txt");  
Scanner inputFile = new Scanner(myFile);
```

Pass the `File` object as an argument to the `Scanner` class constructor.

Reading Data From a File

```
Scanner keyboard = new Scanner(System.in);  
System.out.print("Enter the filename: ");  
String filename = keyboard.nextLine();  
File file = new File(filename);  
Scanner inputFile = new Scanner(file);
```

- The lines above:
 - Creates an instance of the `Scanner` class to read from the keyboard
 - Prompt the user for a filename
 - Get the filename from the user
 - Create an instance of the `File` class to represent the file
 - Create an instance of the `Scanner` class that reads from the file

Reading Data From a File

- Once an instance of `Scanner` is created, data can be read using the same methods that you have used to read keyboard input (`nextLine`, `nextInt`, `nextDouble`, etc).

```
// Open the file.  
File file = new File("Names.txt");  
Scanner inputFile = new Scanner(file);  
// Read a line from the file.  
String str = inputFile.nextLine();  
// Close the file.  
inputFile.close();
```

Exceptions

- The `Scanner` class can throw an `IOException` when a `File` object is passed to its constructor.
- So, we put a `throws IOException` clause in the header of the method that instantiates the `Scanner` class.
- See Example: [ReadFirstLine.java](#)

Detecting The End of a File

- The `Scanner` class's `hasNext()` method will return `true` if another item can be read from the file.

```
// Open the file.
File file = new File(filename);
Scanner inputFile = new Scanner(file);
// Read until the end of the file.
while (inputFile.hasNext())
{
    String str = inputFile.nextLine();
    System.out.println(str);
}
inputFile.close(); // close the file when done.
```

Detecting the End of a File

- See example: [FileReadDemo.java](#)