10.3 Techniques for Any File

- The Class **`File`**
- Programming Example: Reading a File Name from the Keyboard
- Using Path Names
- Methods of the Class **`File`**
- Defining a Method to Open a Stream

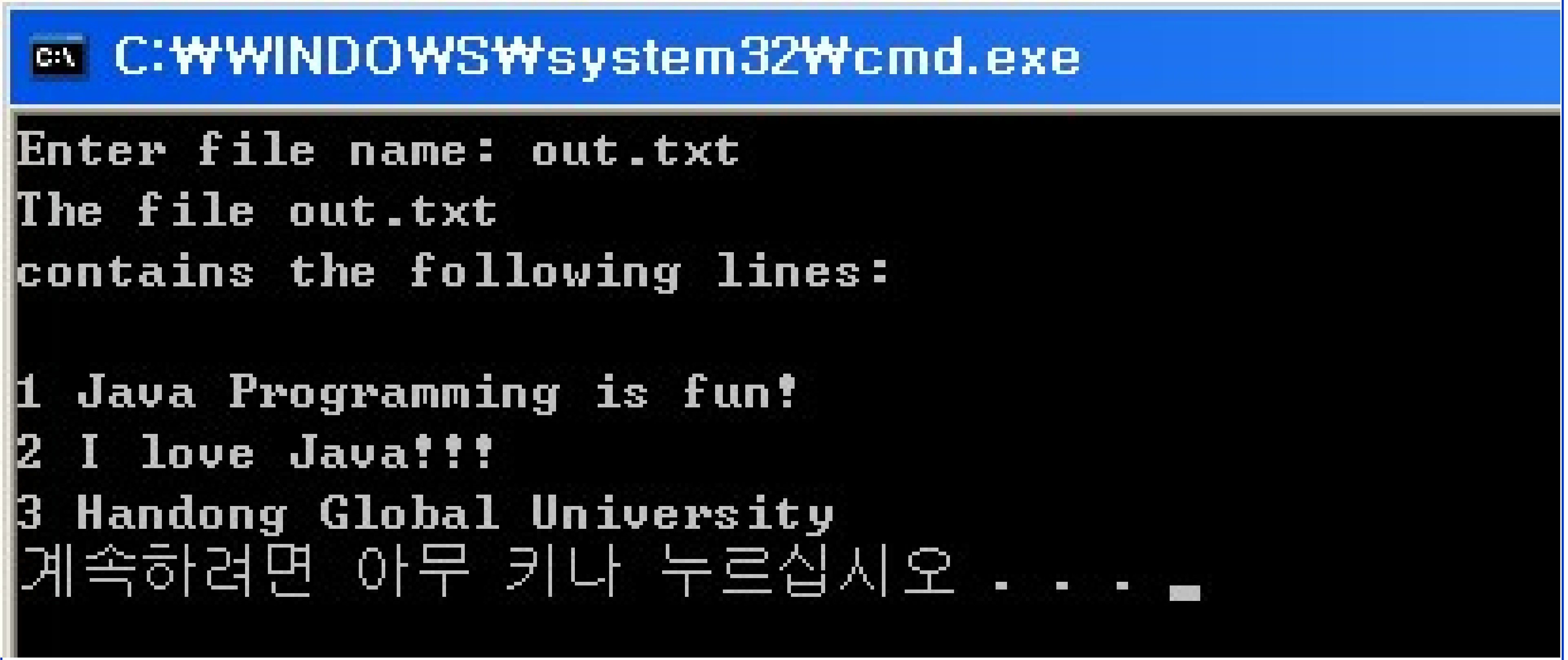
The Class `File`

- Class provides a way to represent file names in a general way
  - » A `File` object represents the name of a file
- The object
  ```
  new File ("treasure.txt")
  ```
  is not simply a string
  - » It is an object that *knows* it is supposed to name a file

- File : Acts like a wrapper class for
- A file name like "`numbers.dat`" has only `String` properties
- But a file name of type `File` has some very useful
  » `exists`: tests to see if a file already exists
  » `canRead`: tests to see if the operating system will let you read a file

```
File fileObject = new File("treasure.txt");

if (! fileObject.exist()) System.out.println("NO file by that name.");
if (! fileObject.canRead()) System.out.println("Not allowed to read from
that file.");
```

```
// Listing 10.3

import java.util.Scanner;
import java.io.File;
import java.io.FileNotFoundException;
public class TextFileInputDemo2
{
    public static void main (String [] args)
    {
        System.out.print ("Enter file name: ");
        Scanner keyboard = new Scanner (System.in);
        String fileName = keyboard.next ();
        Scanner inputStream = null;
        System.out.println ("The file " + fileName + "\n" +
                "contains the following lines:\n");
        try
        {
            inputStream = new Scanner (new File (fileName));
        }
        catch (FileNotFoundException e)
        {
            System.out.println ("Error opening the file " +
                    fileName );
                    System.exit (0);
        }
        while (inputStream.hasNextLine ())
        {
            String line = inputStream.nextLine ();
            System.out.println (line);
        }
        inputStream.close ();
    }
}
```



```
Enter file name: out.txt
The file out.txt
contains the following lines:

1 Java Programming is fun!
2 I love Java!!!
3 Handong Global University
계속하려면 아무 키나 누르십시오 . . .
```

Using Path Names

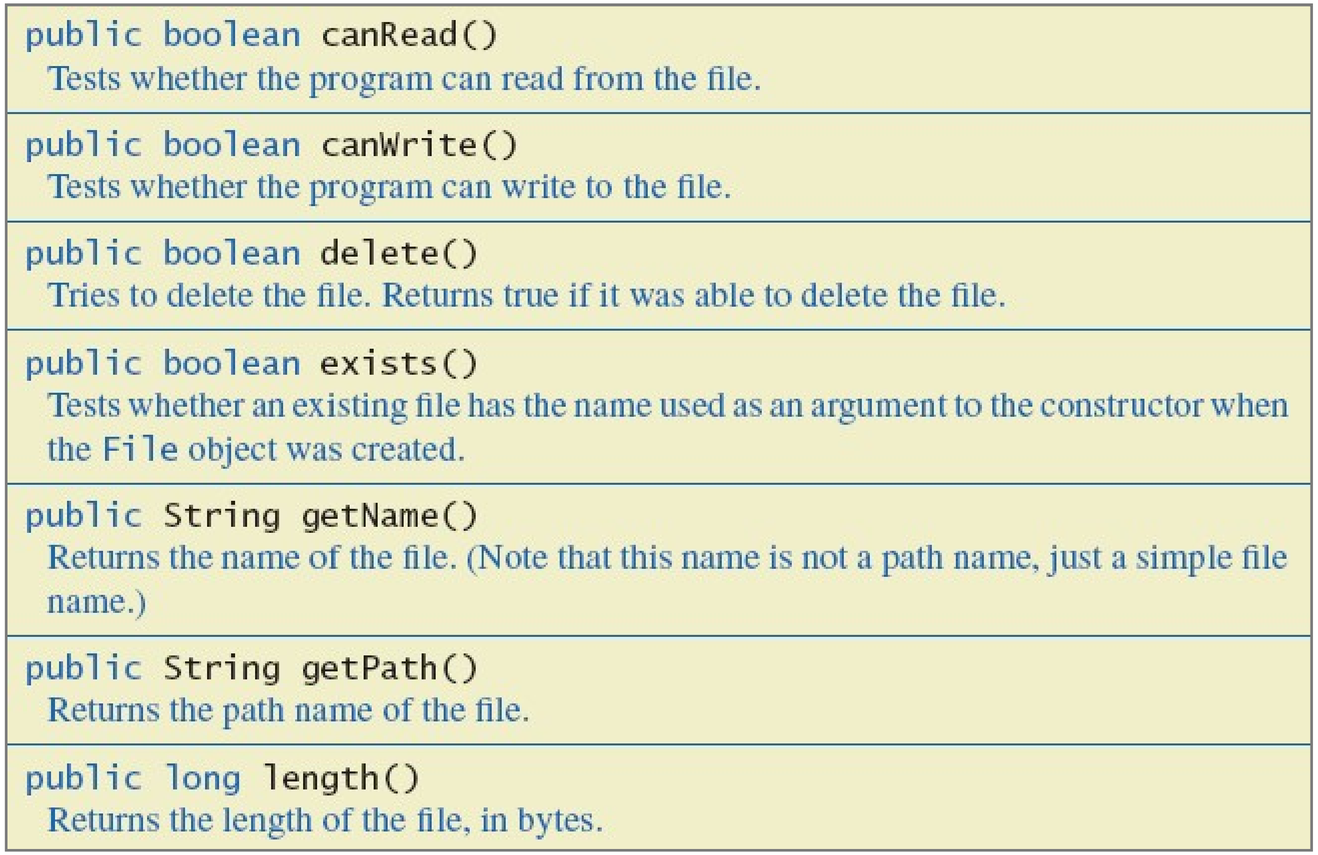
- Files opened in our examples assumed to be in same folder as where program run
- Possible to specify path names
  - » Full path name
  - » Relative path name
- Be aware of differences of pathname styles in different operating systems
  - » Unix : Scanner inputStream = new Scanner(new File(“/user/smith/home.work1/data.txt”));
  - » Windw : Scanner inputStream = new Scanner(new File(“D:\\homework\\hw1\\data1.txt”));

Methods of the Class File

- Recall that a **`File`** object is a system-independent abstraction of file's path name
- Class **`File`** has methods to access information about a path and the files in it
  - » Whether the file exists
  - » Whether it is specified as readable or not
  - » Etc.

Methods of the Class File

- Figure 10.4 Some methods in class **`File`**

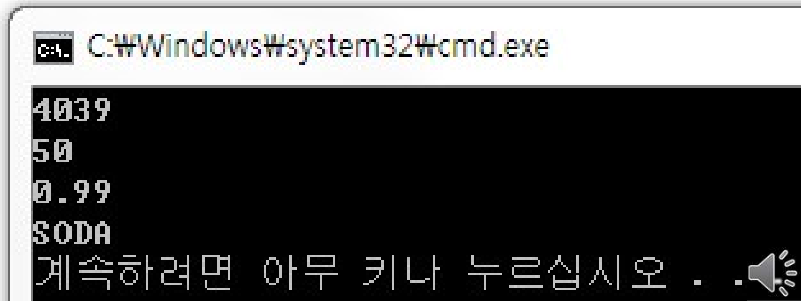


`public boolean canRead()` Tests whether the program can read from the file.
`public boolean canWrite()` Tests whether the program can write to the file.
`public boolean delete()` Tries to delete the file. Returns true if it was able to delete the file.
`public boolean exists()` Tests whether an existing file has the name used as an argument to the constructor when the `File` object was created.
`public String getName()` Returns the name of the file. (Note that this name is not a path name, just a simple file name.)
`public String getPath()` Returns the path name of the file.
`public long length()` Returns the length of the file, in bytes.

Processing a Comma-Separated Values File

- Public String[] split(String delimeter)

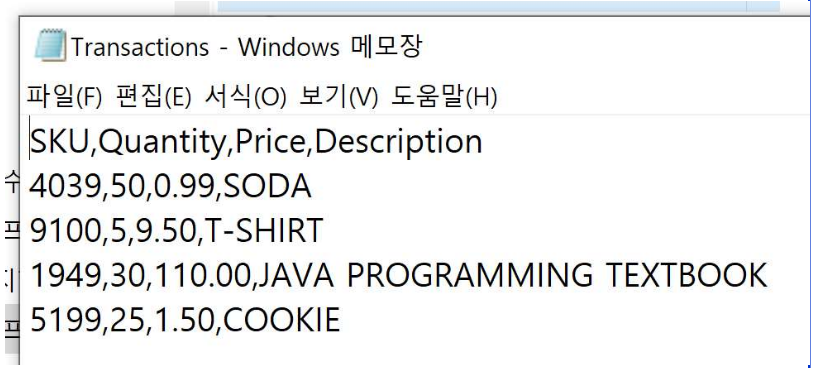
```
import java.util.Scanner;

public class CommaTest
{
    public static void main(String[] args)
    {
        String line = "4039,50,0.99,SODA";

                        String[] ary = line.split(",");
                        System.out.println(ary[0]);
                        System.out.println(ary[1]);
                        System.out.println(ary[2]);
                        System.out.println(ary[3]);

    }
}
```


C:₩Windows₩system32₩cmd.exe

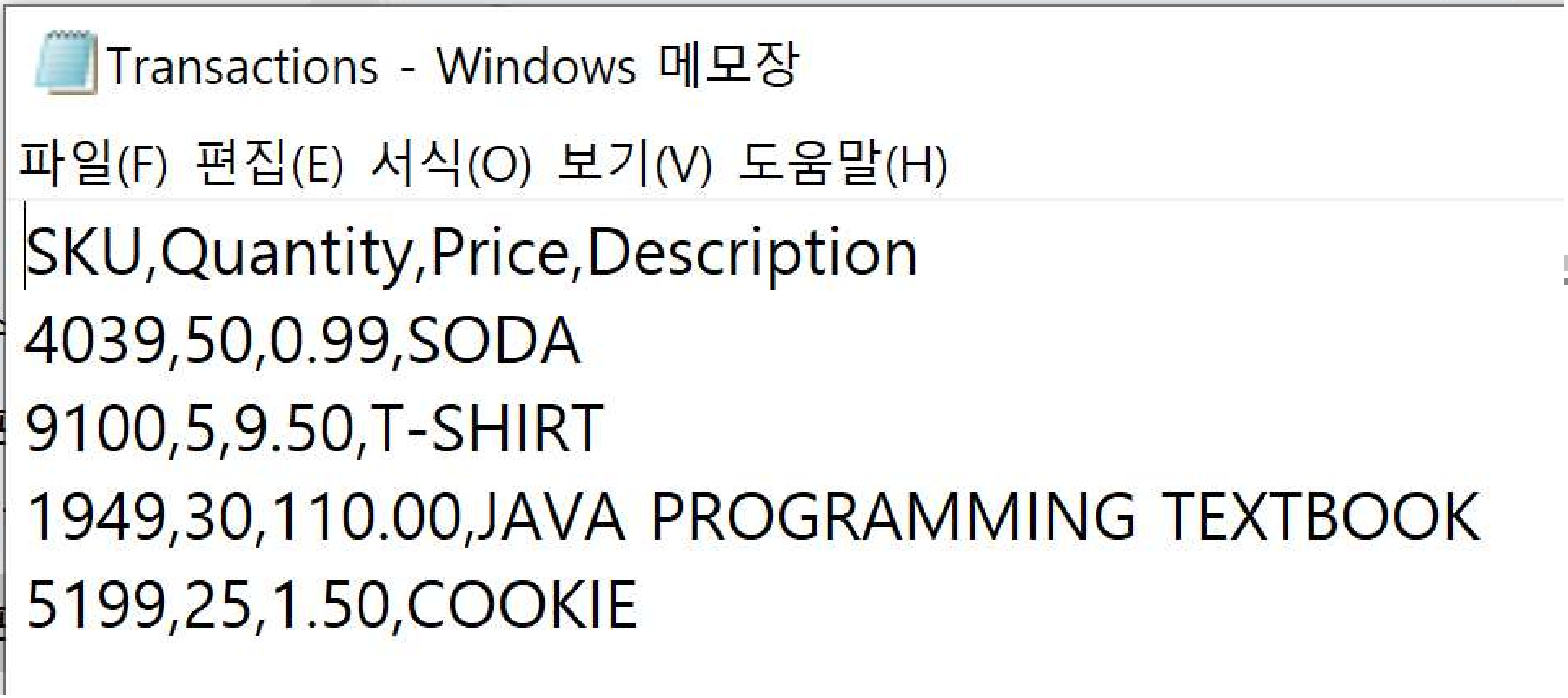
```
4039
50
0.99
SODA
계속하려면 아무 키나 누르십시오 . . .
```

LISTING 10.4 processing a Comma-Seperated Values File Containing Sales Transactions (part 1 of 2)

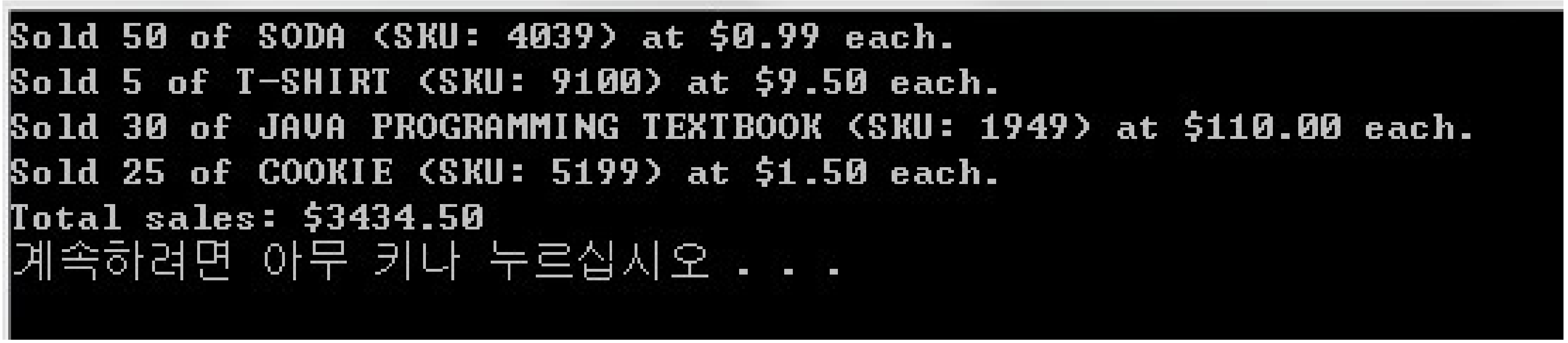
```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.File;
import java.util.Scanner;

public class TransactionReader
{
    public static void main(String[] args)
    {
        String fileName = "Transactions.txt";
        try
        {
            Scanner inputStream = new Scanner(new File(fileName));
            // Read the header line
            String line = inputStream.nextLine();
                                    // Total sales
            double total = 0;
// Read the rest of the file line by line
        while (inputStream.hasNextLine())
        {
           // Contains SKU,Quantity,Price,Description
           line = inputStream.nextLine();
           // Turn the string into an array of strings
           String[] ary = line.split(",");
           // Extract each item
           String SKU = ary[0];
           int quantity = Integer.parseInt(ary[1]);
           double price = Double.parseDouble(ary[2]);
           String description = ary[3];
           // Output item
           System.out.printf("Sold %d of %s (SKU: %s) at $%1.2f each.\n",
                       quantity, description, SKU, price);
           // Compute total
           total += quantity * price;
           }
           System.out.printf("Total sales: $%1.2f\n",total);
        inputStream.close( );
     }
     catch(FileNotFoundException e)
     {
        System.out.println("Cannot find file " + fileName);
     }
     catch(IOException e)
     {
        System.out.println("Problem with input from file " + fileName);
     }
   }
}
```


Transactions - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
SKU,Quantity,Price,Description
4039,50,0.99,SODA
9100,5,9.50,T-SHIRT
1949,30,110.00,JAVA PROGRAMMING TEXTBOOK
5199,25,1.50,COOKIE
```


Transactions - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
SKU,Quantity,Price,Description
4039,50,0.99,SODA
9100,5,9.50,T-SHIRT
1949,30,110.00,JAVA PROGRAMMING TEXTBOOK
5199,25,1.50,COOKIE
```



```
Sold 50 of SODA (SKU: 4039) at $0.99 each.
Sold 5 of T-SHIRT (SKU: 9100) at $9.50 each.
Sold 30 of JAVA PROGRAMMING TEXTBOOK (SKU: 1949) at $110.00 each.
Sold 25 of COOKIE (SKU: 5199) at $1.50 each.
Total sales: $3434.50
계속하려면 아무 키나 누르십시오 . . .
```

끝