MODULE 4

KEYBOARD AND SCREEN I/O

SCANNER – a class in the java.util package. The method of this class are used to read data from the standard input device and store data into variables.

To use the method of Scanner in a computer program, first, import the class using the following statements:

import java.util. Scanner // or java.util.*;

To create a Scanner object and connect it to the System.in object, you write a statement similar to the following:

Scanner input = new Scanner(System.in);

The portion of the statement to the left of the assignment operator, **Scanner input** declares an object of type Scanner.

The portion of the statement to the right of the assignment operator, **new Scanner(System.in),** creates a Scanner object that is connected to the System.in property. In other words, the created Scanner object is connected to the default input device. The keyword new is required by Java; you will use it whenever you create objects that are more complex than the primitive data types.

The Scanner class contains methods that retrieve values from an input device. Each retrieved value is a token, which is a set of characters that is separated from the next set by whitespace.

TABLE 4.1 SCANNER CLASS METHODS

Method	Description
`nextDouble()`	Retrieves input as a `double`
`nextInt()`	Retrieves input as an `int`
`nextLine()`	Retrieves the next line of data and returns it as a `String`
`next()`	Retrieves the next complete token as a `String`
`nextShort()`	Retrieves input as a `short`
`nextByte()`	Retrieves input as a `byte`
`nextFloat()`	Retrieves input as a `float`. Note that when you enter an input value that will be stored as a `float`, you do not type an *F*. The *F* is used only with constants coded within a program.
`nextLong()`	Retrieves input as a `long`. Note that when you enter an input value that will be stored as a `long`, you do not type an *L*. The *L* is used only with constants coded within a program.

The Scanner class does not contain a nextChar() method. To retrieve a single character from the keyboard, you can use the nextLine() method and then use the charAt() method.

Screen Output

RECAP `println`

You can display lines of text using the method `System.out.println`. The items of output can be quoted strings, variables, constants such as numbers, or almost any object you can define in Java.

SYNTAX

```
System.out.println (Output_1 + Output_2 + ... + Output_Last);
```

EXAMPLES

```
System.out.println("Hello out there!");
System.out.println("Area = " + theArea + " square inches");
```

RECAP `println` Versus `print`

`System.out.println` and `System.out.print` are almost the same method. The `println` method advances to a new line *after* it displays its output, whereas the `print` method does not. For example,

```
System.out.print("one");
System.out.print("two");
System.out.println("three");
System.out.print("four");
```

produces the following output:

```
one two three
four
```

The output would look the same whether the last statement involved `print` or `println`. However, since our last statement uses `print`, subsequent output will appear on the same line as `four`.

REFERENCES:

Bart Baesens, Aimée Backiel, Seppe vanden Broucke, (January 2015). Beginning Java® Programming: The Object Oriented Approach.

Joyce Farrell (2016) Java Programming, Eighth Edition.

Walter Savitch, Kenrick Mock, Arup Bhattacharjee and Soumen Mukherjee, (2015) JAVA An Introduction to Problem Solving & Programming (7th Edition).

D.S Malik, Java Programming: From Problem Analysis to Program Design, Fifth Edition