Java Programming

5-1: Basics of Input and Output

Practice Solution

Lesson Objectives:

- Describe the basics of input and output in Java
- Read data from and write data to the console

Vocabulary:

Identify the vocabulary word for each definition below.

	The physical name of a file, or a symbolic link name.
	A type of node at the bottom of a top-down hierarchical (or inverted tree) that has no node below it.
	A file name that maps to another file.
	A top-down single node hierarchy.
	The top most node of a file system hierarchy, also known as a volume name, and used on the Linux operating system.
	The top most node of a file system hierarchy, also known as a volume name, and used on the Windows operating system.
	A hierarchy of elements, starting from a top-most (or root node) and moving down to nodes without any subordinate nodes.
	Either a relative path, which may be some nodes and then a file name, a file name, or an absolute path with a file name as the last element, or leaf node.
	This type of path starts with a logical mount, like C:\ or D:\ in Windows, or a / (forward slash) or combination of a forward slash and one or more node name, as long as its qualified as a mount point.
	A hierarchy where the top-most node is the root and the bottom-most nodes are leaf nodes.
	A path that starts somewhere other than the root node and ends in a file name.
	The top most node of an absolute or relative path.
	A specialized file that points to another absolute or relative file name.

Try It/Solve It:

1. Create a class with a static main that tests the ability to resolve and print a Path:
   - Create an instance of a FileSystem class.
   - Create an instance of the following Path interface.

- Print the constructed Path with System.out.println() method.

2. Identify the main limitations of the Java.io Package.

3. Create a class that does the following:
   - Using a pre-Java 7 solution, create a class that tests streams in the static main.
   - The class should instantiate a new File class, a new FileReader class, and new BufferedReader class.
   - Read lines by using the readLine() method call.
   - The file path used should be: C:/JavaProgramming/employees.txt
   - The file should handle errors when the file is not found as well as reading the contents of the file when it is found.