Java Programming

5-2: Input and Output Fundamentals

Practice Activities

Lesson Objectives:

- Use streams to read and write files
- Read and write objects by using serialization

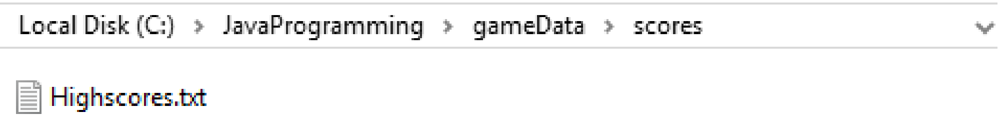
Vocabulary:

Identify the vocabulary word for each definition below.

	The process of converting a stream of data into an object.
	Sequences of bytes or characters transmitted from one program to another program or a file.
	A back slash in front of another character makes Java treat the next character as an ordinary character.
	The process of converting a stream of data into an object.
	Sequences of bytes or characters transmitted from another program or user's activity.
	A file name that maps to another file.
	A path that is direct rather than indirect.
	This is the output stream for error raised by a program, also known as the second channel.
	The join of two related paths, or the remaining part of a join after subtracting part of the path.
	Types of standard input, output, and error; and qualified object types in Java.
	The output for debugging message and ordinary reports and messages.
	Any keyboard, mouse, or touchscreen input to a program.
	The process of converting an object or file to a series of bytes.

Try It/Solve It:

1. Create a class with a static main that tests the ability to resolve and print a Path:
   a) Create an instance of a FileSystem class.
   b) Resolve an instance of a Path interface from a directory and filename path.



Local Disk (C:) › JavaProgramming › gameData › scores

Highscores.txt

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

2. Open the accountgenerator project that was introduced in the practice exercises for section 4_1.

   a) The Employee class should implement the Serializable interface so that it can be Serialized.

   b) To test serialization create a serializeData method in the AccountGenerator class that accepts an Employee object as a parameter and does not return a value. Serialize the object to file named employee.ser in the project folder.

   c) To test de-serialization create a deSerialize method in the AccountGenerator class that deoes not accept any parameters and returns an Object value. The deSerialize method should throw a classNotFoundException. The object should be read from the employee.ser file in the project folder.

   d) To test de-serialization create a deSerialize method in the AccountGenerator class that deoes not accept any parameters and returns an Object value. The deSerialize method should throw a classNotFoundException. The object should be read from the employee.ser file in the project folder.