



Java I/O Fundamentals

Oracle® Certified Professional, Java® SE 7 Programmer
Module 7

Module 7 - Objectives

- + Read and write data from the console.
- + Use streams to read and write files.

I/O

- + I/O is broken down into two types in Java: byte and character.
- + Character streams are handled by Reader/Writer classes.
- + Byte streams are handled by InputStream/OutputStream classes.
- + Reader, Writer, InputStream, and OutputStream are abstract classes.
- + The Decorator pattern is used to provide additional functionality to the I/O base classes.

I/O (cont.)

- + Most I/O classes are wrapped to combine capabilities.
- + A `FileReader` is often wrapped with a `BufferedReader` to improve the efficiency of the I/O operation.
- + A `FileWriter` is often wrapped with a `BufferedWriter` to improve its operation.

I/O (cont.)

- + Prior to Java 5 `FileReader` and `FileWriter` did not have constructors for `File` and `String`. The addition of these constructors made use of the classes easier.

Reading Input from a Console

- + The most common method used for many years for reading console data is using an `InputStream` wrapped around `System.in`.
- + Code generated prior to Java 6 used this technique, and it is fairly well understood and tested.
- + Since Java 6 the `Console` object is used.

Console

- + A new class in Java 6.
- + Console represents the display and keyboard, or DOS/UNIX terminal.
- + It is obtained using `System.console();`
- + The Console makes it easier to accept input from the command line.

File (NIO.2)

- + This is new in Java 7.
- + The `Files`, `Paths`, and `Path` classes can make your life with file handling much easier.
- + `Path` represents an abstraction of a file system path.
- + The `Path` object is an interface, and concrete implementations are returned using the `Paths` object.
- + Since `File` already exists in the API, the `Files` class was created as a enhancement for with additional functionality.

Files (NIO.2)

- + The `Files` class is the way to go when creating, copying, or modifying files on a modern operating system.
- + The `Files` class provides a number of convenience methods to manipulate files.
- + This should be your first API choice when looking for file manipulation.



Attribution-NonCommercial-ShareAlike 3.0 Unported

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit
<http://creativecommons.org/licenses/by-nc-sa/3.0/>.