Rex
Jones II

(Transcript)
Java Output

Introduction

Java programs perform I/O through streams. I/O stands for Input Output. You know how we print to the console using System.out.println. That's part of Java's I/O system to produce information. We can also use the I/O system to consume information.

There are 3 streams defined in the java.io package to read data or write data. The streams are Byte, Character, and Predefined.

1. Byte is defined by 2 class hierarchies: InputStream and OutputStream
2. Character is also defined by 2 class hierarchies but those classes are Reader and Writer. Reader is used for input and Writer is used for output.

This is a list of the Byte Streams and Character Streams. The original Java version only included the Byte Stream. However, the Character Stream was created to parallel the corresponding Byte stream. This list came from Oracle's site.

3. Number 3 is the Predefined Stream containing the System class with 3 fields: in, out, and err. System.in refers to the standard input which defaults to the keyboard. System.out refers to the standard output which defaults to the console. System.err refers to the standard error which also defaults to the console just like System.in.

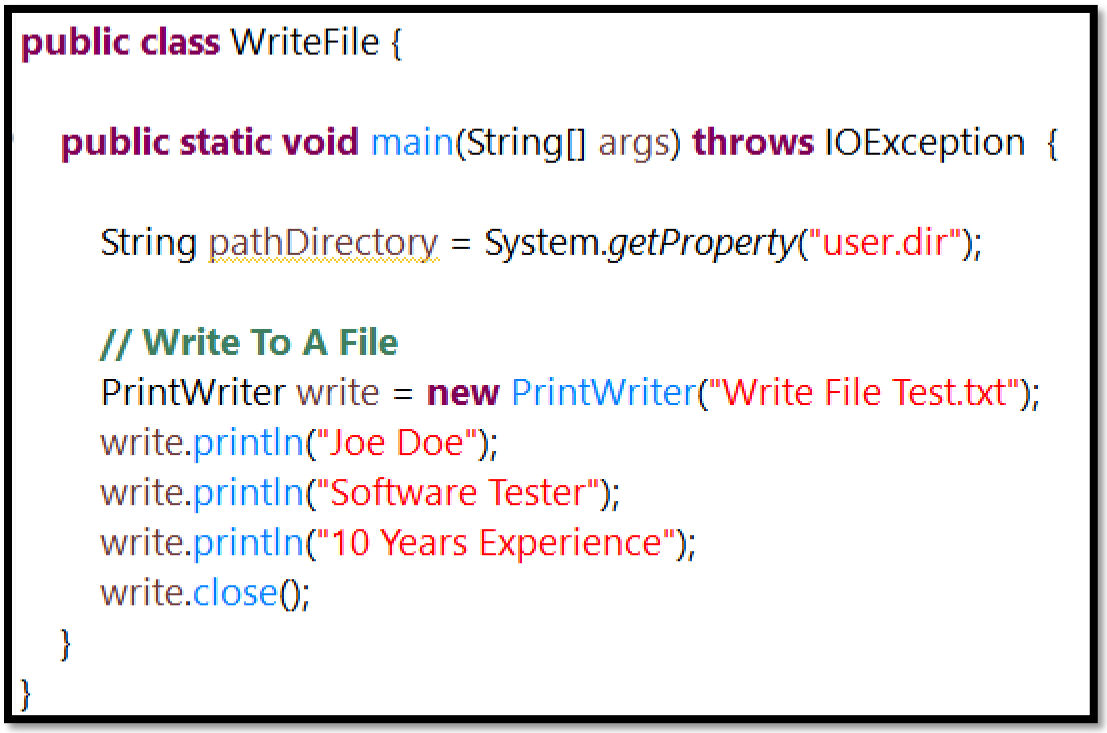
We are going to focus on the Character Streams that was designed to handle input and output of characters.

Java Output

This session will cover Java's Output. For starters, let's print the current path directory. String pathDirectory = System.getProperty("user.dir").
sysout(pathDirectory + "\n") and Run. We see the directory in the Console. System.out is an object of type PrintStream which contains print() and println() that contain aspects of the run-time environment.

Now, it's time to // Write To A File. PrintWriter is a class that allows us to print on the file, object write = new PrintWriter().We see some constructors. File is the one we need. Which file are we printing to file? "Write File Test.txt". This will be the filename. Add a throws declaration of IOException.

System.out is very similar to PrintWriter. Look what happens when I type write.pri. Do you see the print statements? Select println(""). The difference is System.out prints to the console and PrinterWriter prints to a file. First line is "Joe Doe". Second Line ("Software Engineer"), and last will be ("10 Years Experience").We must always close at the end write.close().



```
public class WriteFile {

  public static void main(String[] args) throws IOException  {

    String pathDirectory = System.getProperty("user.dir");

    // Write To A File
    PrintWriter write = new PrintWriter("Write File Test.txt");
    write.println("Joe Doe");
    write.println("Software Tester");
    write.println("10 Years Experience");
    write.close();
  }
}
```

Close releases the file for other system resources in the Operating System. Open Package Explorer and Run. We see Write File Test in the Package Explorer. Open the file and the information is correct. The purpose of this video was to give you an example of how to handle output characters in Java. Next, I will show you how to handle input characters.