Input Output Streams Package: java.io

By,

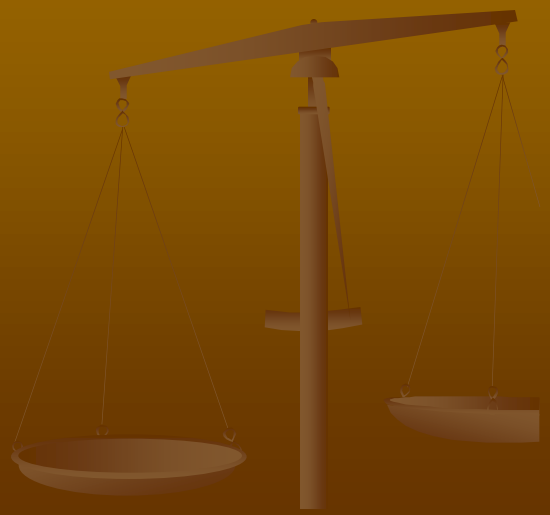
Hitha Paulson

Assistant Professor, Dept. of Computer Science

LF College, Guruvayoor

Basics

- How to transfer data between two entities??
- Stream based data transfer
- If one of the entity is an I/O device, It is an I/O operation
- It happens from
  - Console: Less frequently used
  - File
  - Forms: widely used
  - Network



Streams

- **Stream**: is an Abstraction that either produces or consumes information
- A stream is linked to a physical device by the Java I/O system
- Streams handle data irrespective of the attached device
- Streams are handled by using the classes and methods defined in java.io package

Byte/Character Streams

- Java defines two types of streams,
- Byte Stream
  - Streams in the form of bytes.
  - Applies to binary data
- Character Stream
  - Streams in the form of collection of Characters
  - Handle data in Unicode format
  - Sometimes more efficient than Byte streams
  - At the lowest level, data moving in byte format

Stream Classes

- Byte Stream Classes
  - Top level classes: InputStream, OutputStream
  - Both are abstract classes and contain abstract methods read() and write()
  - Large number of subclasses are derived to support different devices, such as Disk, Network, ...
- Character Stream Classes
  - Top level classes: Reader, Writer (handles UNICODE character streams)
  - Both are abstract classes and contain abstract methods read() and write()

Byte Stream Classes

- BufferedInputStream
- ByteArrayInputStream
- DataInputStream
- FileInputStream
- FilterInputStream
- ObjectInputStream
- PipedInputStream
- PrintStream
- RandomAccessFile

Character Stream Classes

- BufferedReader
- CharArrayReader
- FileReader
- FilterReader
- InputStreamReader
- LineNumberReader
- PipedReader
- PrintWriter
- StringReader

Predefined Streams

- java.lang.System class contains three predefined stream variables
  - **in, out, err**
  - These variables are public, static and final
  - System.in :- Standard input stream object representing keyboard. Object of InputStream class
  - System.out :- Standard output stream object of PrintStream class
  - System.err :- Represents standard error stream and object of PrintStream class

Reading Console Input

- Character streams are more preferred than bytes streams
- Class used to read character stream from keyboard
  - BufferedReader: Takes a Reader as its parameter
  - InputStreamReader: Class used to convert bytes into character. Takes System.in as its parameter

Writing Console Output

- Byte Oriented
  - PrintStream derived from ***OutputStream***
  - void write(int byteval), writes the specified byte of data into the associated stream
  - Eg: System.out.write('A');
- Character Oriented
  - PrintWriter is a character based class for console output in international standard
  - PrintWriter object is initialized using any of the ***OutputStream*** object

File Streams

- All files are Byte oriented
- File Stream classes
  - ***FileInputStream***:- to read data from file
  - ***FileOutputStream***:- to write data into the file
  - Both classes takes "filename" as argument in the constructor and **open** the file
  - Both classes throws ***FileNotFoundException*** when file not found or file cannot be created
  - An existing file will destroy, when a file having same name is opened by using FileOutputStream

Read/Write Data

- read() method
  - defined with FileInputStream can be used to read a single byte
  - Read() method returns -1, when EOF is encountered
  - It throws IOException
- write() method
  - Defined with FileOutputStream used to write one byte of data into the File
  - It throws IOException
- close() method
  - Used to close the stream created with FileInputStream and FileOutputStream
  - It also throws IOException