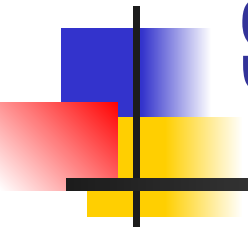
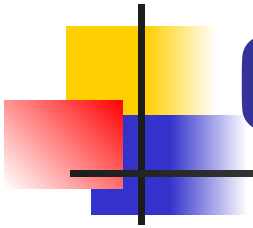


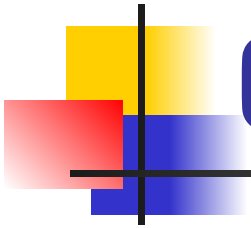
Session

I/O Streams



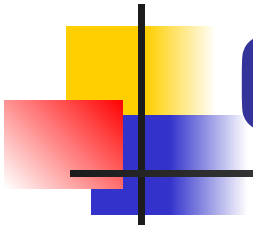
Objectives

- Define streams
- Describe the InputStream and OutputStream classes
- Describe the I/O Byte array



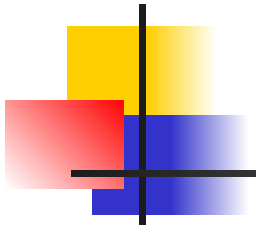
Objectives

- Describe the File I/O Classes
- Filtered Input and Output Classes
- Buffered I/O Classes
- Reader and Writer Classes
- Character Array and String Input and Output Classes
- PrinterWriter Class

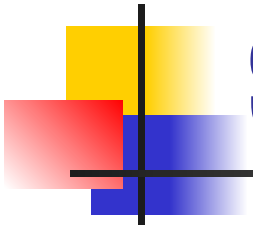


Objectives

- Describe DataInput Interface
- Describe DataOutput Interface
- Describe RandomAccessFile Class
- Explain java.awt.print Package

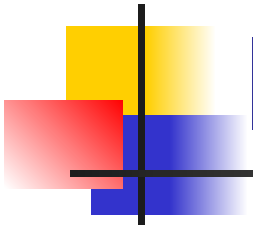


- http://www.studytonight.com/java/java-io-stream.php
-
- http://www.javatpoint.com/java-io



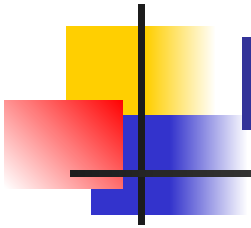
Streams

- Streams are pipelines for sending and receiving information in Java programs
- When a stream is read or written, the other threads are blocked
- If an error occurs while reading or writing a stream, an IOException is thrown
- Class ‘java.lang.System’ defines the standard input and output streams



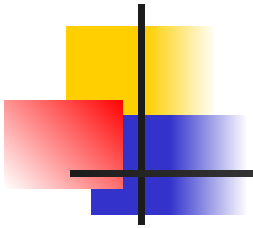
I/O Stream classes

- System.out class
- System.in class
- System.err class



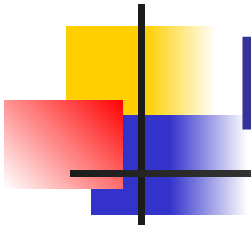
InputStream Class

- Is an abstract class
- Defines how data is received
- Provides a number of methods for reading, and taking streams of data as input
- Methods:
  - **read( )**
  - **available( )**
  - **close ( )**
  - **mark ( )**
  - **markSupported( )**
  - **reset( )**
  - **skip( )**



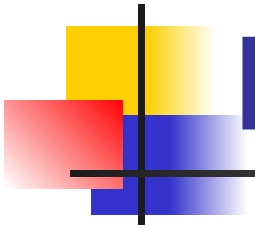
OutputStream Class

- Is abstract
- Defines how outputs are written to streams
- Provides a set of methods that help in creating, writing and processing output streams
- Methods
  - **write(int)**
  - **write(byte[ ])**
  - **write(byte[ ], int, int)**
  - **flush( )**
  - **close( )**



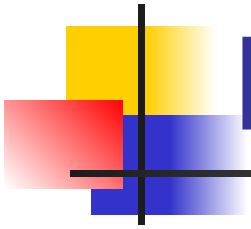
Byte Array Input

- Use memory buffers
- **ByteArrayInputStream** Class
  - Creates an input stream from the memory buffer which is an array of bytes
  - Does not support any new methods
  - Overrides methods of the class InputStream, such as ‘read()’, ‘skip()’, ‘available()’ and ‘reset()’



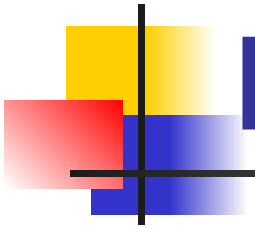
Byte Array Output

- Use memory buffers
- **ByteArrayOutputStream** Class
  - Creates an output stream on a byte array
  - Provides additional capabilities for the output array to grow, to accommodate the new data that is written
  - Also provides methods to convert the stream to a byte array, or a String object



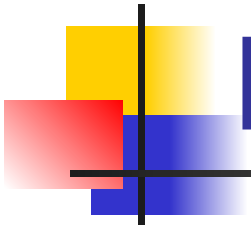
Byte Array Output (Contd...)

- Methods of **ByteArrayOutputStream** Class:
  - **reset( )**
  - **size( )**
  - **writeTo( )**



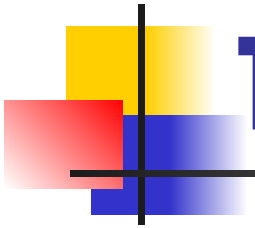
File I/O classes

- Classes that help Java to support file input and output operations:
  - File
  - FileDescriptor
  - FileInputStream
  - FileOutputStream
- File, FileDescriptor, and RandomAccessFile classes are used to support direct or random access input and output



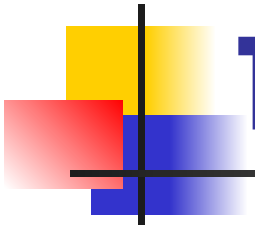
File Class

- Is used to access file and directory objects
- The files are named according to the file-naming conventions of the host operating system
- This class provides constructors for creating files and directories
- All common file and directory operations are performed using the access methods and directory methods that the File class provides



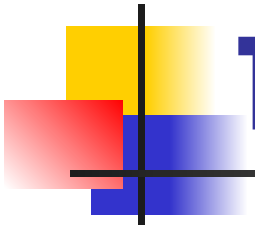
The FileDescriptor Class

- Provides access to the file descriptors
- Does not provide any visibility to specific information that the operating system maintains
- Provides only one method, called 'valid( )'



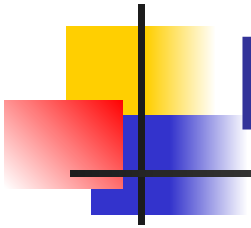
The FileInputStream Class

- Allows the input to be read from a file in the form of a stream
- Objects are created using a filename String, File, or a FileDescriptor object, as an argument
- Overrides the methods of the InputStream class; it also provides the 'finalize( )' and 'getFD( )' methods



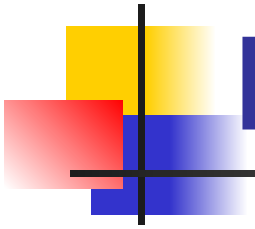
The FileOutputStream Class

- Allows output to be written to a file stream
- Objects are also created using a filename String, File, or a FileDescriptor object, as an argument
- This class overrides the methods of the OutputStream class, and provides the 'finalize( )' and 'getFD( )' methods



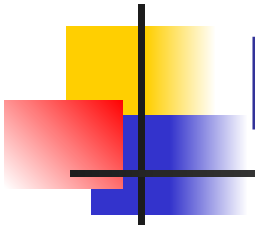
Filtered Input and Output

- A filter(s):
  - Is a type of stream that modifies the way an existing stream is handled
  - Are basically used to adapt streams to specific program needs
  - Sits between an input and an output stream
  - Performs some special processing on the bytes transferred from input to output
  - Can be combined to perform a sequence of filtering options



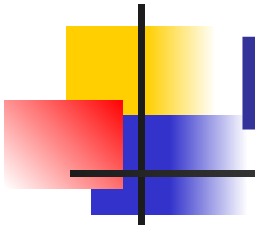
Filtered Input and Output (Contd...)

- The filtered input and output streams classes that Java provides help filter the I/O in a number of ways



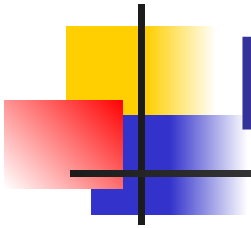
FilterInputStream Class

- Is an abstract class
- Is the parent of all filtered input stream classes
- Provides the capability to create one stream from another
- One stream can be read and provided as the output to another stream
- Maintains a separate object of the class 'InputStream'
- Allows the creation of multiple chained filters



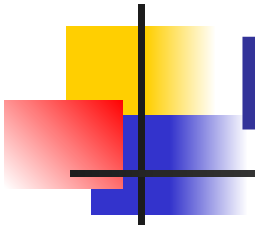
FilterOutputStream Class

- Is the supplement to the class 'FilterInputStream'
- Is the parent of all filtered output stream classes
- Maintains the object of the class 'OutputStream' as an 'out' variable
- Data written to this class can be modified to perform filtering operations, and then forwarded to the 'OutputStream' object



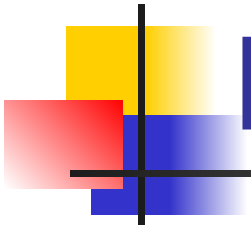
Buffered I/O

- A buffer:
  - Is a storage place for data
  - Can provide data instead of going back to the original source of the data
- Java uses buffered input and output to temporarily cache data that is read from, or written to, a stream
- While performing buffered input:
  - A large number of bytes are read at a time, and stored in an input buffer
  - When the program reads the input stream, the input bytes are read from the input buffer



Buffered I/O (Contd...)

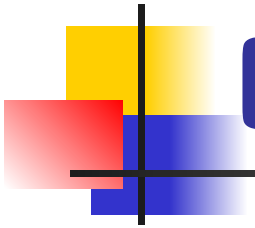
- In case of output buffer, a program writes data to a stream
- Output data is stored in an output buffer
- Data is stored until the buffer becomes full, or the output stream is flushed
- Finally, the buffered output is forwarded to the destination of the output stream



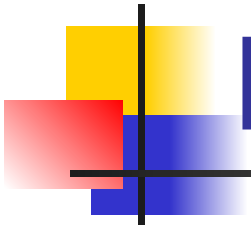
BufferedInputStream Class

- Automatically creates and maintains a buffer to support input buffering
- As the class 'BufferedInputStream' is a filter, it can be applied to certain objects of the class 'InputStream'
- Can also be combined with other input files
- Uses several variables to implement input buffering

BufferedInputStream Class (Contd...)

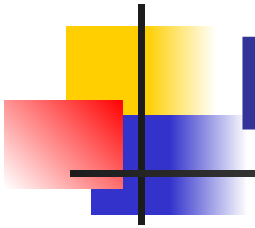


- Defines two constructors:
  - One allows the size of an input buffer to be specified
  - Other does not
- Both constructors take an object of the class 'InputStream' as an argument
- Overrides the access methods that the InputStream provides, and does not introduce any new methods



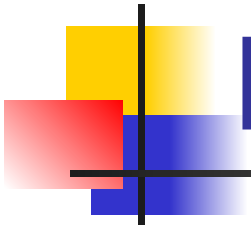
BufferedOutputStream Class

- Performs output buffering in a manner that corresponds to the class 'BufferedInputStream'.
- Defines two constructors. It allows us to specify the size of the output buffer in a constructor, as well as provide a default buffer size
- Overrides all the methods of the 'OutputStream' class, and does not introduce any methods.



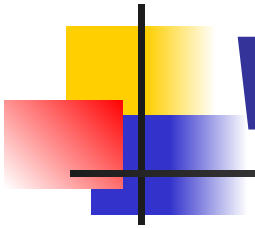
Reader and Writer Classes

- Are abstract classes
- Are at the top of the class hierarchy that supports the reading and writing of unicode character streams



Reader Class

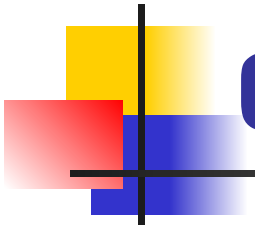
- Supports the following methods:
  - **read( )**
  - **reset( )**
  - **skip( )**
  - **mark( )**
  - **markSupported( )**
  - **close( )**
  - **ready( )**



Writer Class

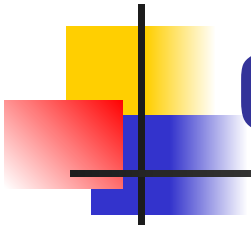
- Supports the following methods:
  - **write( )**
  - **flush( )**
  - **close( )**

Character Array Input and Output classes



- Supports input and output from memory buffers
- Support 8-bit character input and output
- 'CharArrayReader' does not add any new methods to the ones that the class 'Reader' provides

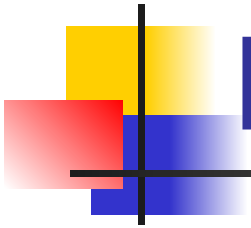
Character Array Input and Output classes



- 'CharArrayWriter' class adds the following methods to the ones that the class 'Writer' provides:
  - **reset( )**
  - **size( )**
  - **toCharArray( )**
  - **toString( )**
  - **writeTo( )**

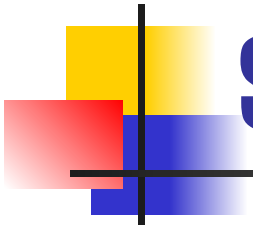
Character String Input and Output classes

- The class 'StringReader' helps read the character input from a string
- It does not add any methods to those that the class Reader provides
- The class 'StringWriter' helps to write character output to a 'StringBuffer' object
- This class adds the following methods:
  - **getBuffer( )**
  - **toString( )**



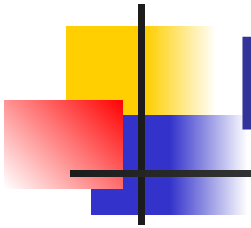
PrinterWriter Class

- Implements an output
- Has additional methods, which help print the basic types of data
- The PrintWriter is a character-oriented version of the 'PrintStream'
- Is an improvement upon the 'PrintStream' class; this class uses a platform-dependent line separator to print lines, instead of the '\n' character
- Provides better support than PrintStream for Unicode characters
- Methods:
  - **checkError( )**
  - **setError( )**



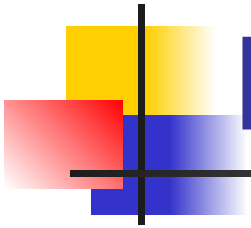
Summary

- File I/O classes
- Buffered I/O classes
- Reader and Writer classes
- Character Array and String I/O classes
- PrinterWriter class



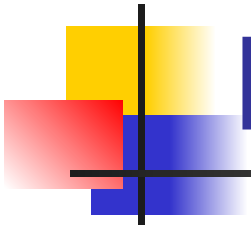
DataInput Interface

- Is used to read bytes from a binary stream, and reconstruct data in any of the Java primitive types
- Allows us to convert data from the Java modified UTF-8 format to the String form
- Defines a number of methods, including methods for reading Java primitive data types



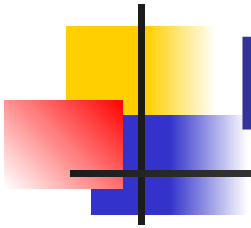
DataInput Interface Methods

- boolean readBoolean( )
- byte readByte( )
- char readChar( )
- short readShort( )
- long readLong( )
- float readFloat( )
- int readInt( )
- double readDouble( )
- String readUTF( )
- String readLine( )



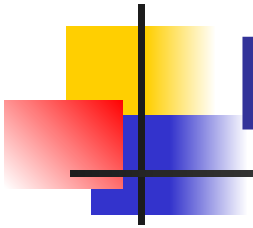
DataOutput Interface

- Is used to reconstruct the data in any of the Java primitive types into a series of bytes
- Writes these bytes onto a binary stream
- Allows us to convert a String into the Java modified UTF-8 format, and write it into a stream
- Defines a number of methods and all methods throw an IOException in case of an error



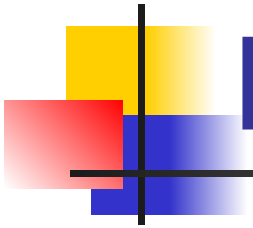
DataOutput Interface Methods

- void writeBoolean(boolean b)
- void writeByte(int value)
- void writeChar(int value)
- void writeShort(int value)
- void writeLong(long value)
- void writeFloat(float value)
- void writeInt(int value)
- void writeDouble(double value)
- void writeUTF(String value)



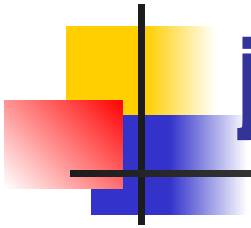
RandomAccessFile Class

- Provides the capability to perform I/O to specific locations within a file
- Data can be read or written to random locations within a file, instead of a continuous storage of information
- 'seek( )' method supports random access
- Implements both data input and output
- Supports basic file read/write permissions
- Inherits methods from the 'DataInput' and 'DataOutput' classes



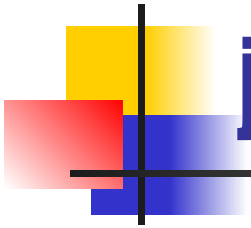
RandomAccessFile Class Methods

- seek( )
- getFilePointer( )
- length( )



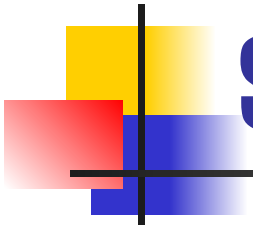
java.awt.print Package

- Consists of the following interfaces
  - Pageable:
    - Defines the methods used for objects representing the pages to be printed
    - Specifies the number of pages to be printed, and whether the current page, or a range of pages is to be printed
  - Printable:
    - Specifies the 'print( )' method used for printing a page on a 'Graphics' object
  - PrinterGraphics:
    - Provides access to the 'PrinterJob' object



java.awt.print Package (Contd...)

- 'PrinterGraphics' interface provides the following classes:
  - Paper
  - Book
  - PageFormat
  - PrinterJob
- 'java.awt.print' package throws the following exceptions:
  - PrinterException
  - PrinterIOException
  - PrinterAbortException



Summary

- DataInput and DataOutput interfaces
- RandomAccessFile class
- ‘jawa.awt.print’ package