Course Name:
Advanced Java

Lecture 9
Topics to be covered

- I/O streams
  - Byte Streams
  - Character Streams
  - InputStream
  - ByteArrayInputStream
  - FileInputStream
  - FilterInputStream

Introduction

- A Stream is a sequence of bytes or characters that travel from a source to a destination over a communication path.
- The two end points of a stream represent the source and destination of data.

OR

- A Stream an object that either delivers data to its destination (screen, file, etc.) or that takes data from a source (keyboard, file, etc.)
- It acts as a buffer between the data source and destination

Cont..

- **Input stream**: a stream that provides input to a program
  - System.in is an input stream
- **Output stream**: a stream that accepts output from a program
  - System.out is an output stream
- A stream connects a program to an I/O object
  - System.out connects a program to the screen
  - System.in connects a program to the keyboard

Cont…

- The java.io package includes various streams to handle input/output from various devices.
- Based on the type of data transferred through them, the streams in java can be broadly classified into two categories:

1. Byte Streams
2. Character Streams

Byte Streams

- You can use a byte stream to handle the input and output of bytes in an application.
- Based on the direction of flow of data, byte streams can further be divided into two sub-categories:

- Input Streams:- An input stream is an object that an application uses to read data from a data source, such as an existing file or an input device.

- Output Streams:- An output stream is used by an application to write data to a destination, which can be a file or a console.

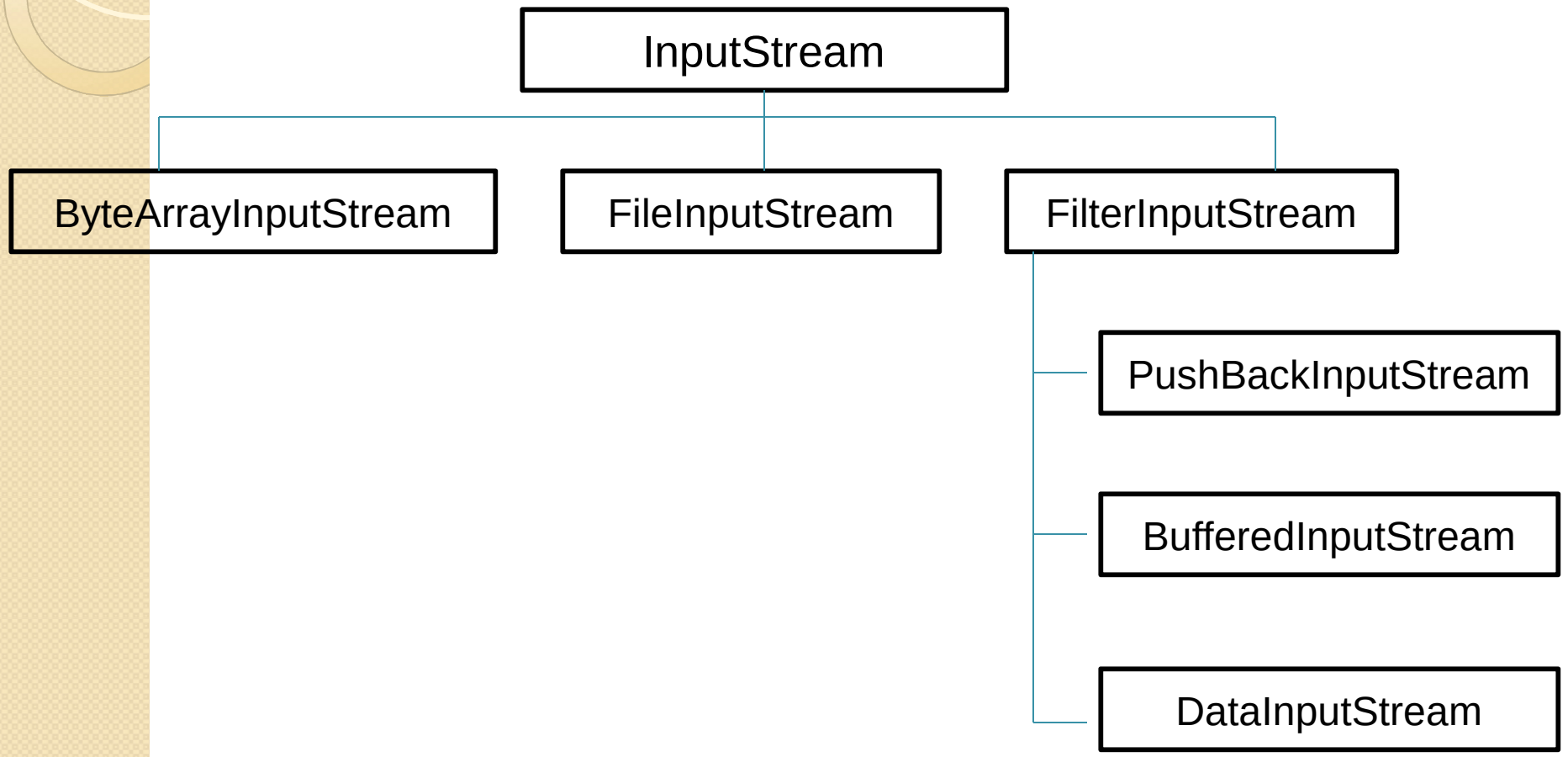
Character Streams

- A character stream adds a level of abstraction to streams by enabling an application to read or write data as a stream of characters instead of bytes.

Input Streams

- Input streams enable an application to read bytes of data from various input devices.
- To the application, the input stream acts similar to a source of data.
- The java.io package includes a range of classes that can be used to create and manage input streams.
- InputStream class is the superclass for all the input stream classes.
- It is an abstract class that defines the basic functionality required for data input.

InputStream class Hierarchy



InputStream class methods

- The InputStream class defines a range of methods that you can use to accept data from an input stream.
- These methods are inherited by the subclasses in the hierarchy when they extend the InputStream class.

There are several methods in this class:

- read()
- read(byte b[])
- int read(byte b[], int off, int len)
- int available()
- void close()

Cont…

Method	Explanation
int read()	Reads a byte of data from the input stream. The return value specifies the integer representation of the next available byte. The method returns -1 if the end of file is encountered.
int read(byte b[])	Reads bytes of data from the input stream and stores them in an array. The return value specifies the number of bytes read, or -1 if the end of file is encountered.
int read(byte b[],int off, int len)	Reads the number of bytes specified by the third argument from the input stream and stores it in an array. The second argument specifies the starting position of the bytes within the array. The return value specifies the number of bytes read or -1 if the end of file is encountered.

Cont…

Method	Explanation
int available()	Returns the number of bytes available for reading from the input stream.
Void close()	Closes an input stream and releases all the resources associated with it.

Markable Streams

- Sometimes, you may not want to extract the complete data from a stream. In such situations, you can use bookmarks that specify the starting point for reading a stream.
- There are certain streams in java.io package that provide the capability of placing bookmarks on the stream and resetting them. These streams are known as markable streams.
- These streams can be assigned a bookmark and read from the marked position.
- If a stream is marked, it must have some memory associated with it to keep track of the data between the marked position and the current position of the stream.

Methods in Markable stream

Method	Description
boolean markSupported()	Returns true if the stream supports a bookmark
void mark(int readlimit)	Marks a position on the stream and identifies the number of bytes that can be read before the mark becomes invalid.
void reset()	Repositions the stream to its last marked position.
long skip(long nBytes)	Skips a specific number of bytes in a stream

ByteArrayInputStream

- The ByteArrayInputStream class creates an input stream by using an array of bytes in the memory buffer as the data source.
- To create an input stream, the ByteArrayInputStream class defines the following constructors:

ByteArrayInputStream(byte buff[ ])

- This constructor creates an object of the ByteArrayInputStream class that uses an array of bytes in memory as the data source. The byte array is specified as a parameter in the constructor.

Cont…

- ByteArrayInputStream(byte buff[], int start, int len)

  This constructor creates an object of the ByteArrayInputStream class that uses a subset of an array of bytes in memory as the data source.

- The first parameter specifies the byte array .
- The second parameter specifies the starting index of the subset within the array.
- The third parameter specifies the length of the subset from the starting index.

FileInputStream

- The FileInputStream class is used to create a stream that reads bytes of data from an existing file.
- To create an input stream, the FileInputStream class defines the following constructors:

FileInputStream(String fPath)

This constructor creates an object of the FileInputStream class that uses a file as the data source. The complete path of the source is passed as a parameter in the constructor.

Cont…

FileInputStream(File fObj)

This constructor creates a FileInputStream object that uses a file as the data source. The parameter specified in the constructor represents the source file specified as an object of the File class.

- The FileInputStream class overrides the methods of the InputStream class.
- You can read a single byte or an array of bytes using the read() method.

FilterInputStream

- The FilterInputStream class defines the basic functionality to read bytes from an underlying input stream.
- In addition, it provides a filter to extract the required data from the source stream. It then passes the filtered data to the underlying input stream.

It has three subclasses:

- BufferedInputStream
- PushBackInputStream
- DataInputStream