# 0301502 ADVANCED JAVA

| UNIT | MODULES | WEIGHTAGE |
|------|---------|-----------|
| 1 | File Handling | 20 % |
| 2 | Java Collection Framework | 20 % |
| 3 | Event Handling, Swing and GUI Components | 20 % |
| 4 | Swing, GUI Components and Layout Manager | 20 % |
| 5 | Database Connectivity (JDBC) | 20 % |

# INTERNAL EVALUATION

INTERNAL EVALUATION

ASSIGNMENTS

ATTENDANCE

MCQ TEST

5 Assignments
* 5 Marks
= 25 Marks

25 Marks

20 Marks * 5
= 100Marks

# TEXT BOOK

- Java Programming
  - Publisher : - Pearson
  - Author : - Hari Mohan Pandey

## Recommended Book

- Proggramming in Java2
  - Publisher : - A Jaico Boo
  - Author : -  Dr. k. Somasundaram

# UNIT -1 File Handling

- I/O Stream

- The File Class

- Byte Stream

- Disk File Handling

- Filterred Byte Stream

- Sequence Input Stream

- Object Input / Output Stream

- Random Access File

# UNIT -1 File Handling - "I/0 Stream"

- Java program perform Input or Output operation **through Stream.**

- A **stream is a sequence of bytes or characters** that travel from source to destination over a communication path.

- The **java.io package** has plenty of Stream classes.

- There are two fundamentally different way to store data.
  - **Binary format**
  - **Text format**

# UNIT -1 File Handling - "I/0 Stream"

- For Binary format I/O **Byte Streams** claases are used. Byte streams provide the foundation for **8-bits** I/O.

- For Text format I/O **character Streams** classes are used. Character streams provide the foundation for **16-bits Unicode character I/O.**

# UNIT -1 File Handling - "I/0 Stream"

- Data items are made available in **binary format**, we have to use the following classes or subclasses.

  – **InputStream**

  – **OutputStream**

- If data items are available in **text format**, we have to use the following cleasses or subclasses.

  – **Reader**

  – **Writer**

# UNIT -1 File Handling - "I/0 Stream"

```
                    ┌──────────────────────┐
                    │  Java Stream Classes  │
                    └──────────────────────┘
                   ┌───────────────┴───────────────┐
        ┌──────────────────────┐        ┌──────────────────────────┐
        │  Byte Stream Classes  │        │ Character Stream Classes  │
        └──────────────────────┘        └──────────────────────────┘
           ┌───────┴───────┐                 ┌───────┴───────┐
  ┌──────────────┐  ┌──────────────┐   ┌──────────┐   ┌──────────┐
  │ Input Stream │  │Output Stream │   │  Reader  │   │  Writer  │
  │   Classes    │  │   Classes    │   │ Classes  │   │ Classes  │
  └──────────────┘  └──────────────┘   └──────────┘   └──────────┘
  ┌──────────────┐  ┌──────────────┐
  │ Used for Reading│ Used for Write│
  │  from Source   │ │  to Source   │
  └──────────────┘  └──────────────┘
```
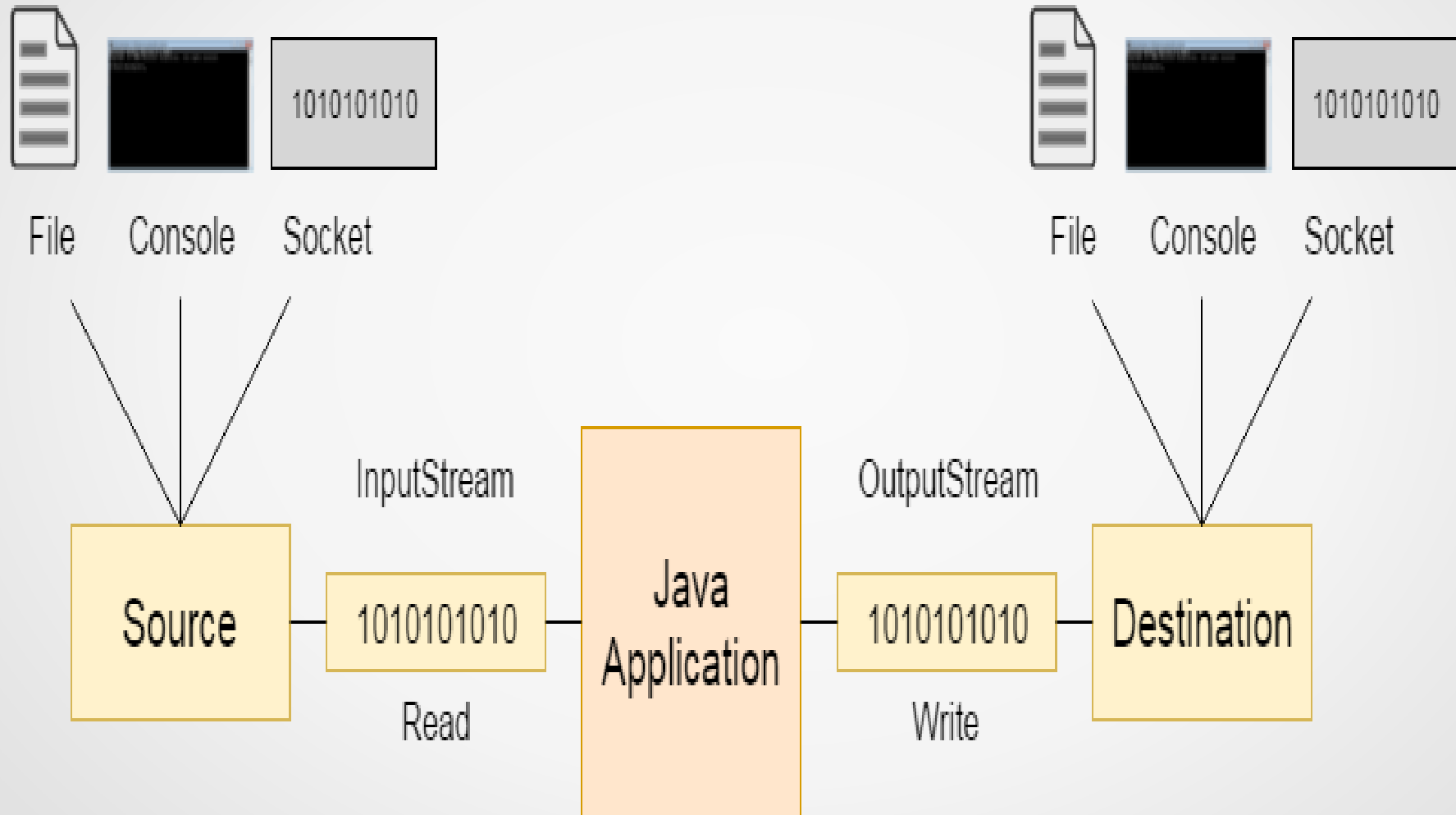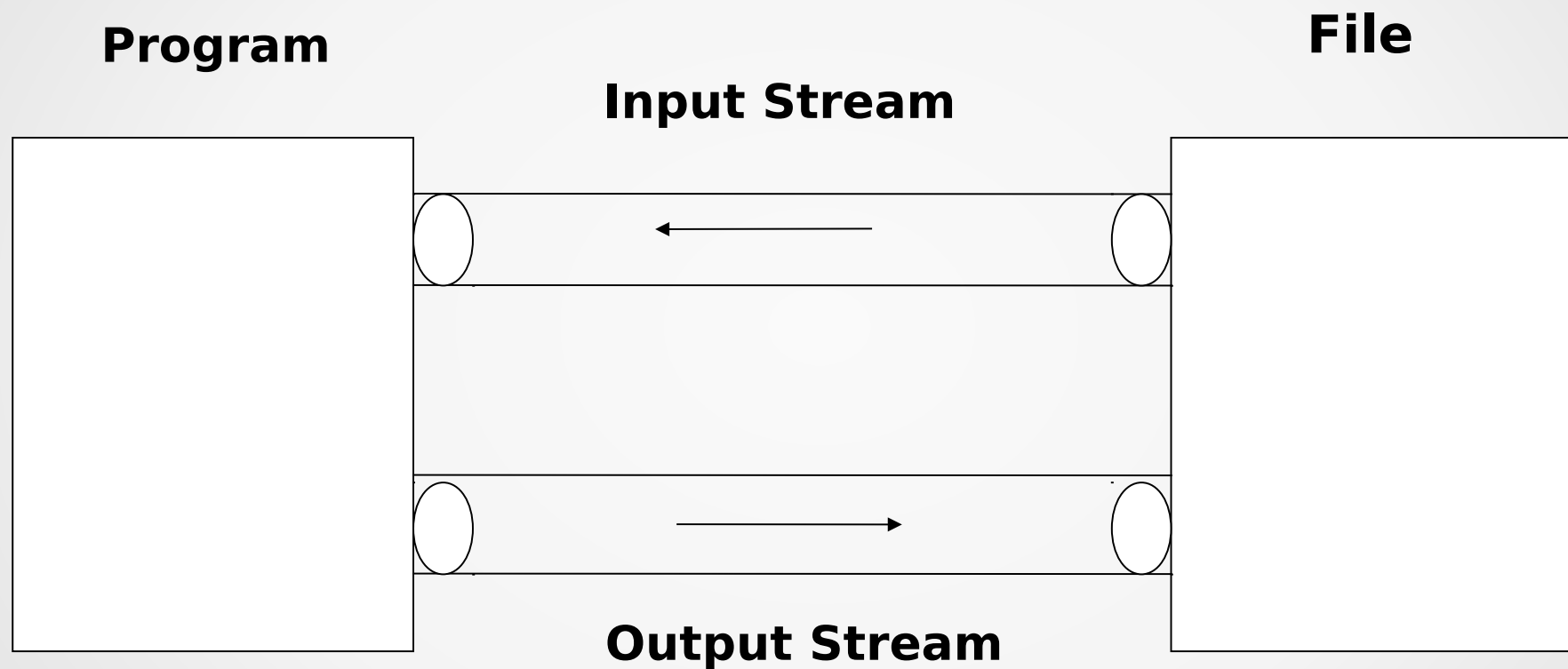
# UNIT -1 File Handling - "I/0 Stream"

- **Stream are two type:**

  - **Input Stream** – The sources from **data read**

  - **Output Stream** – The  destination to **data write**

# UNIT -1 File Handling - "I/0 Stream"

**Program**

**File**

**Input Stream**

**Output Stream**

# UNIT -1 The File Class

- File class is used to **know detail of the particular File.**

- Using this class we **cannot read or write from / to file.**

- File class has follow constructor:

  - *File(String path)*

  - *File(String path, String FileName)*

  - *File(File object, String FileName)*

- *File class has total  17 Methods*

# UNIT -1 File Class Methods

| Method | Purpose of the Method |
|--------|----------------------|
| *String getName()* | Return the **name of the directory or file.** |
| *boolean exists()* | Return **true if director or file exists** else return false |
| *boolean canWrite()* | Return **true if file object is writable** else false. |
| *boolean canRead()* | Return **true if file object is readable** else false. |
| *boolean isDirectory()* | Return **true if it is directory** else false. |

# UNIT -1 File Class Methods

| Method | Purpose of the Method |
|---|---|
| *Boolean isFile()* | Return **true if it is file** else false. |
| *long length()* | Return **the size of file** in bytes. |
| *Boolean renameTo(File newName)* | **Rename the file with newName and return true** if it is successful else false. |
| *boolean delete()* | **Delete the file and return true if** it is successful else false. |
| *String[] list()* | Return **string array of the file names** of the directory. |

# UNIT -1 File Class Methods

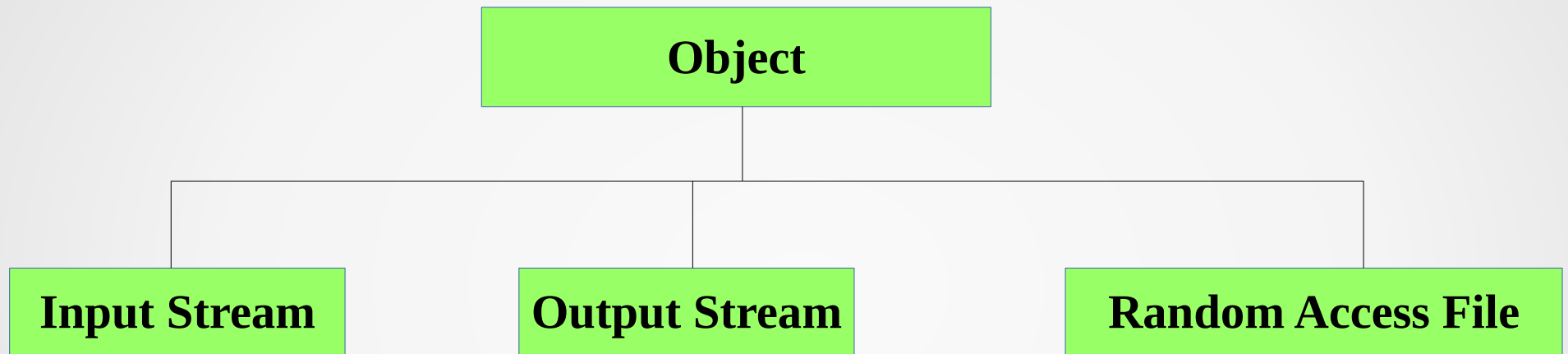| Method | Purpose |
|---|---|
| *String getPath()* | Return **the path** of the invoking object |
| *String getParent()* | Return the **name of parent directory.** |
| *String getAbsolutePath()* | Return the **absolute path** of the invoking object |
| *Boolean isHidden()* | Return **true if the File object is hidden**, otherwise false |
| *Boolean setReadOnly()* | **Sets the file** of the invoking object to **read only mode** , return true on successful setting other wise false |
| *Void deleteOnExit()* | **Deletes the file** of the invoking object upon the **termination of Java Virtual Machine.** |

# UNIT -1 File Class
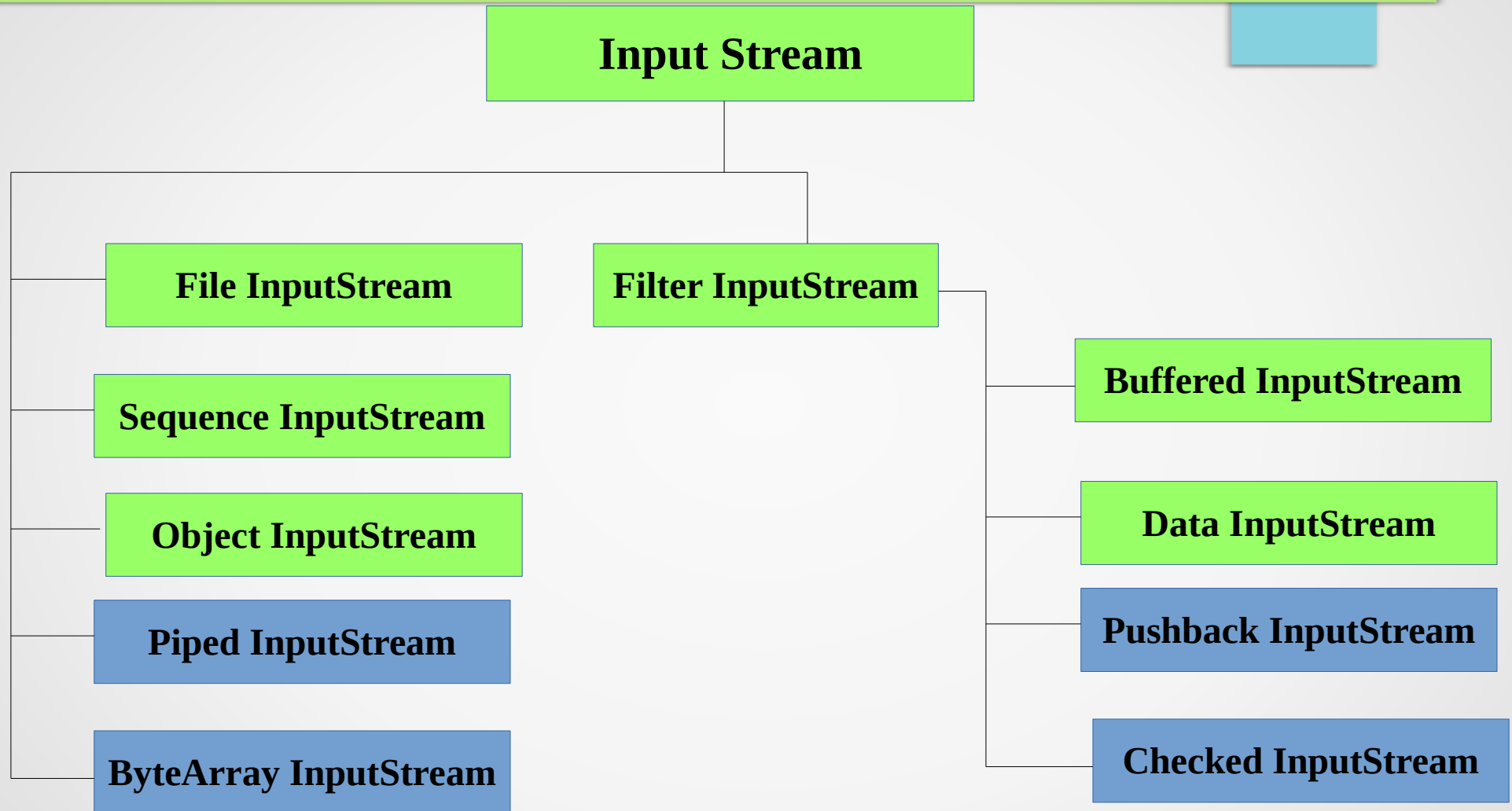
- FileDemo.java

# UNIT -1 I/O Stream

- I/O Operation of **Byte Stream** are handle by

  – InputStream class (Abstract Class)

  – OutputStream class (Abstract Class)

- I/O Operation of **Character Stream** are handle by

  – Reader class (Abstract Class)

  – Writer class (Abstract Class)

- The I/O related classed are available in **java.io package**

# UNIT -1 I/O Stream

```
                        ┌──────────────┐
                        │    Object    │
                        └──────┬───────┘
           ┌───────────────────┼───────────────────┐
   ┌───────────────┐   ┌───────────────┐   ┌─────────────────────┐
   │ Input Stream  │   │ Output Stream │   │ Random Access File  │
   └───────────────┘   └───────────────┘   └─────────────────────┘
```

# UNIT -1 I/O Stream

```
                    ┌─────────────────┐
                    │  Input Stream   │
                    └─────────────────┘
```

**Input Stream**

**File InputStream**

**Filter InputStream**

**Sequence InputStream**

**Buffered InputStream**

**Object InputStream**

**Data InputStream**

**Piped InputStream**

**Pushback InputStream**

**ByteArray InputStream**

**Checked InputStream**

# UNIT -1 I/O Stream

```
                          Output Stream
                               |
        +----------------------+----------------------+
        |                                             |
  File OutputStream                           Filter OutputStream
                                                      |
  Object OutputStream                          Buffered OutputStream
                                                      |
  Piped OutputStream                           Data OutputStream
                                                      |
  ByteArray OutputStream                       PrintStream
                                                      |
                                               Checked OutputStream
```
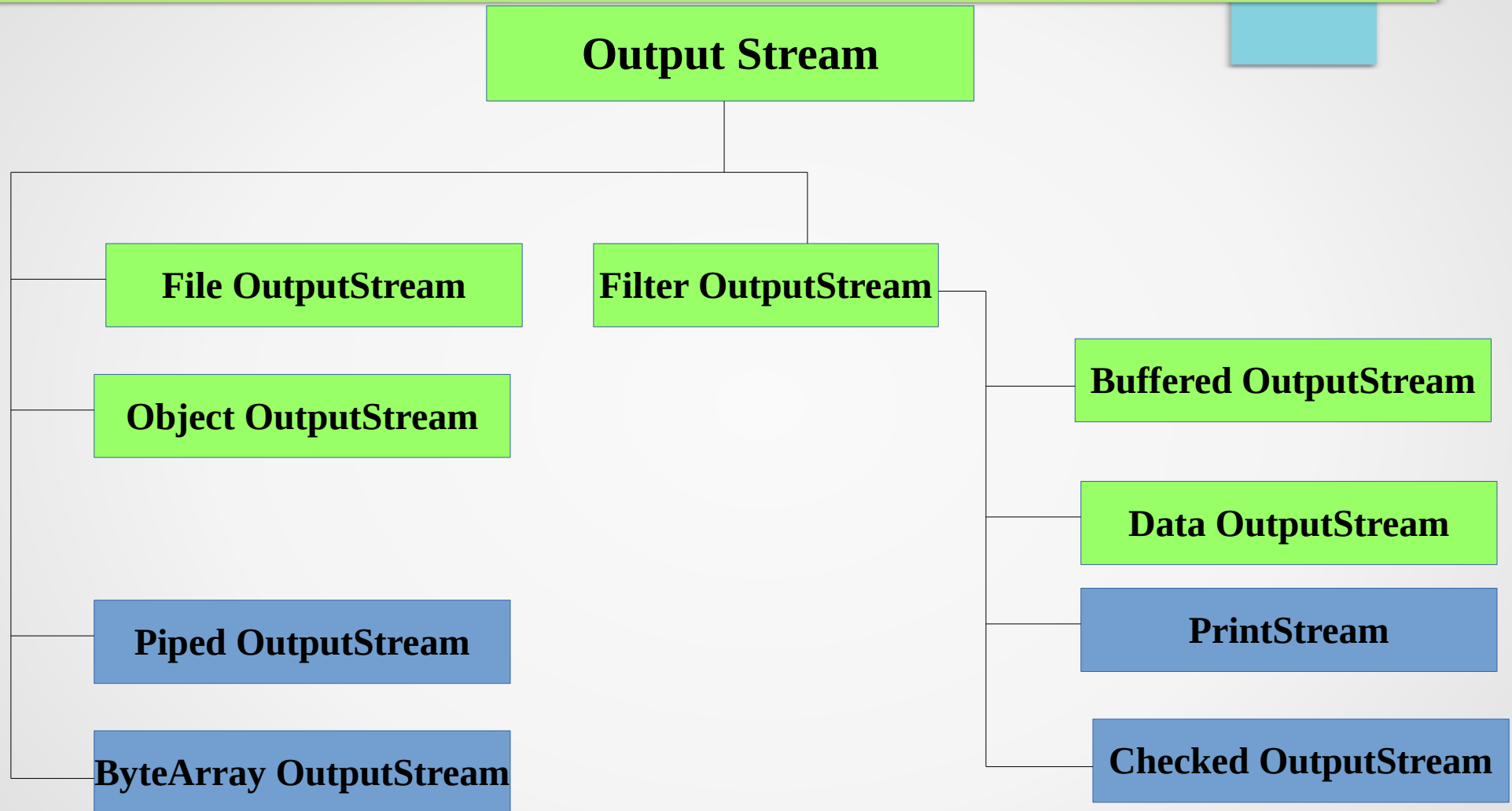
# UNIT -1 Byte Stream

- I/O Operation of **Byte Stream** are handle by

  - **InputStream class (Abstract Class)**

    - This stream, accesses the data as a **sequence of Bytes**. All types other than character or text are dealth in this stream.

    - This class is **Abstract class,** so object of this class **cannot be created object**. But object reference of this type can be declared.

# UNIT -1 Byte Stream- Methods of Input Stream Class

| Method | Purpose of the Method |
|---|---|
| *abstract int read( )* | **Reads one byte** and returns an int representation of that byte; returns -1 at the end of the system |
| *int read (byte ba[])* | **Reads into an array of bytes** 'ba' and returns the number of b.length bytes returns -1 at the end of the stream |
| *int read (byte ba[], int end, int length)* | **Reads length number of bytes into the byte array** 'ba', starting at ba[end] and returns the number of bytes read; returns -1 at the end of the stream |
| *void close( )* | **Closes** the input stream |
| *void mark(int marklen)* | **Puts a mark at the current position** and remembers it until marklen bytes are read |

# UNIT -1 Byte Stream- Methods of Input Stream Class

| Method | Purpose of the Method |
|---|---|
| *boolean marksupported( )* | Returns **true if the stream supports mark** () |
| *void reset ( )* | The **read control is reset to the previous set mark.** |
| *int available ( )* | Return **the number of bytes available for reading** |
| *long skip(long m)* | **Skips 'm' bytes of the input stream** and returns the actual number of bytes skipped |

# UNIT -1 Byte Stream- Methods of Output Stream Class

| Method | Purpose |
|---|---|
| *abstract void write(int b)* | **Writes a single byte** b to an output stream |
| *void write (byte b[])* | **Writes a array of byte** b to an output stream |
| *void write (byte b[], int off, int len)* | **Writes len number of bytes** starting from b[off] to an output stream |
| *void close ( )* | **Closes the output stream** |
| *void flush( )* | **Flushes the output buffer;** that is sends any buffered byte to its destination |

# UNIT -1 Disk File Handling

| Input Stream |
| :---: |

| Output Stream |
| :---: |

| File Input Stream |
| :---: |

| File Output Stream |
| :---: |

# UNIT -1 Disk File Handling

- **FileInput Stream Class**

    - This class is used to read bytes from a disk file.

    - FileInput Stream class has follow constructor:

        - *FileInputStream(String fileName)*

        - *FileInputStream(File Object)*

    - *Both the Constructors can throw*
      ***FileNotFoundException***

## UNIT -1 Disk File Handling File Reading using File Input Stream

- Readbyte.java

# UNIT -1 Disk File Handling

- **FileOutput Stream Class**

  - This class is used to write bytes into a disk file.

  - FileInput Stream class has follow constructor:

    - *FileOutputStream(String fileName)*

    - *FileOutputStream(File Object)*

    - *FileOutputStream(String fileName, boolean append)*

  - *All the Constructors can throw **IOException** or a **SecurityException***

# Output Stream Class

- FileWriteDemo.java

- ReadAndWrite.java

- File_read_write.java

# Practics

- Create a java programme in which, Read data from keyboard and write it into binary file.

- Create a java programme, it read data from two different file and mearge into third binary file.

# UNIT -1 Filtered Byte Stream

- The basic **byte streams access** the data in **byte form.**

- For **converting bytes to useful forms** such as char, string, int etc. **Java hava several streams**.

- These **streams work on other streams and convert between byte to useful form** or vice versa.

- **Such Streams that can take other stream as argument are called filtered streams**.

- The java has

    - *FilterInputStream class*

    - *FilterOutputStream class*

# UNIT -1 Filtered Byte Stream
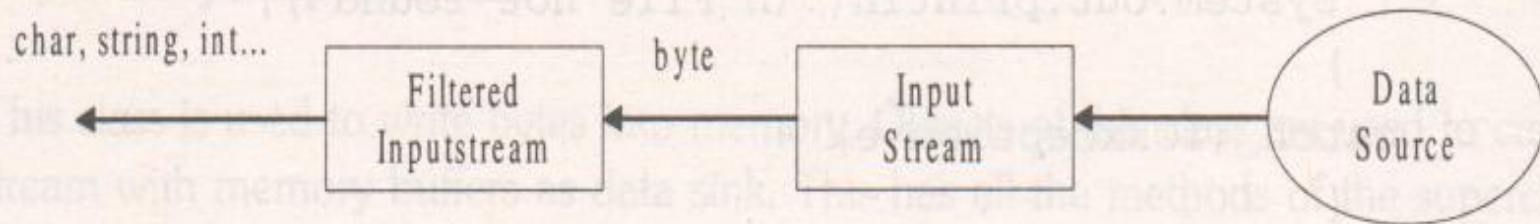


a. FilteredInputStream
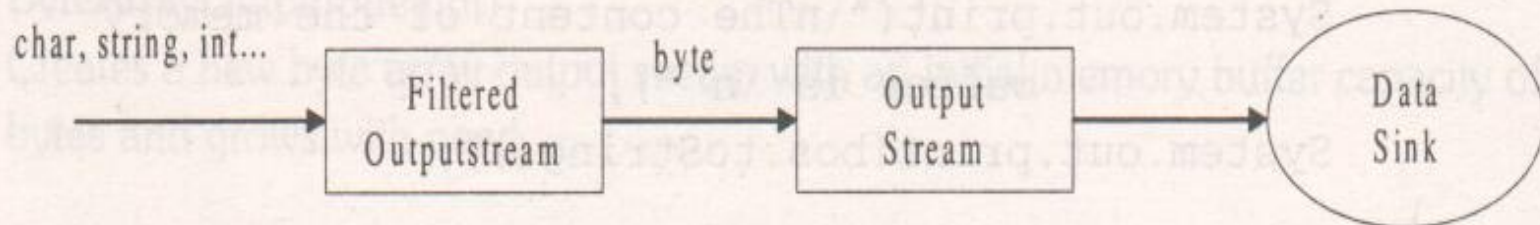
b. FilteredOutputStream

# UNIT -1 BufferedInputStream

- ***BufferedInputStream* is a subclass** of FilterInputStream class.

- With the use of this class byte can be read and store in a buffer memory before sending it to the destination, **once the buffer is full, the bytes can be sent to the destination.**

- BufferInputStream class has follow constructor:

  - BufferedInputStream(InputStream ins);

  - BufferedInputStream(InputStream ins, int size);

**Bufinput.java**

# UNIT -1 BufferedOutputStream

- *BufferedOutputStream* **is a subclass** of FilterOutputStream class.

- With the use of this class byte can be write and store in a buffer memory before sending it to the destination, **once the buffer is full, the bytes can be sent to the destination.**

- BufferedOutputStream class has follow constructor:

  - BufferedOutputStream(OutputStream ous);

  - BufferedOutputStream(OutputStream ous, int size);

**Bufwrite.java**

# UNIT -1 DataInputStream Class

- It is a **subclass of FilterInputStream class.**

- This class **helps to convert the bytes from an underlying stream to basic type char, byte, int etc.**

- DataInputStream class has follow constructor:

    – DataInputStream(Inputstream ins)

# Methods of DataInputStream Class

| Method | Purpose |
|---|---|
| *int read (byte[] b)* | **Read bytes into the byte array b** returns the numbers of bytes read; **returns -1 at the end** of the stream. |
| *int read (byte[] b, int off, int len)* | **Read bytes** into the byte array b starting at b[off], len number of bytes, **returns the total numbers of bytes read**, returns -1 at the end of the file. |
| *void readFully(byte[] b)* | **Reads bytes** into the byte array b, **the maximum bytes read being the capacity of the byte array.** |
| *Boolean readBoolean()* | **Reads a boolean** value from the input stream |
| *Byte readByte()* | **Reads an 8bit signed byte** from the input stream |

# Methods of DataInputStream Class

| Method | Purpose |
|---|---|
| *short redShort()* | **Reads two bytes** from the input stream and **treats it as 16-bit int** |
| *char readInt()* | **Read two bytes** from the inut stream and **treats its as a 16 bit unicode character.** |
| *long readLong()* | **Read 8 bytes from the** input stream and **treats it as a long** |
| *float readFloat()* | **Reads 4 bytes** of the input stream and **treats it as a float** |
| *void readFully(byte[] b, int off, int len)* | **Read bytes** into b, starting at b[off], len number of bytes |

# Methods of DataInputStream Class

| Method | Purpose |
| --- | --- |
| *int skipBytes(int n)* | **Skips n bytes** and **returns the actual number of bytes skipped.** |
| *int readInt()* | **Read 4 bytes** from the input stream and **treats it as an int.** |
| *double readDouble()* | **Read 8 bytes** from the input stream and **treat it as a double.** |

# UNIT -1 DataOutputStream Class

- It is a **subclass of FilterOutputStream class.**

- This class **helps to convert the basic type to bytes** and pass it to an underlaying output stream.

- Constructure :

  – DataOutputStream(OutputStream ous)

# Methods of DataOutputStream Class

| Method | Purpose |
|---|---|
| *void write(int b)* | **Write the lower 8 bits** of b to the underlying output stream |
| *void write(byte bf[], int off, int len)* | **Write len bytes from the byte** array bf, starting at bf[off] to the underlying output stream. |
| *void writeBoolean(boolean b)* | **Write the boolean** value b to the underlying output stream as 1 byte value. |
| *void writeByte(int in)* | **Write the int value** as byte value to the underlying stream. |
| *void writeShort(int in)* | **Write a short** to the underlying output stream as two bytes, the high byte is written first. |

# Methods of DataOutputStream Class

| Method | Purpose |
|--------|---------|
| *void writeChar(int in)* | **Write a char as 2 byte** value to the underlying stream. |
| *void writeInt(int in)* | **Write an int as 4 bytes** to the underlying stream. |
| *void writeLong(long ln)* | **Write a long as 8 bytes** to the underlying stream. |
| *void writeFloat(float f)* | **Converts the float to an int using a special metod and writes the int as 4 bytes** to the underlying stream. |
| *void writeDouble(double d)* | **Converts the double to a long using a special method and writes the long as 8 bytes** to the underlying output stream |

# Methods of DataOutputStream Class

| Method | Purpose |
|---|---|
| *void writeBytes(String s)* | **Write the String s as a sequence of bytes** to the underlying ouput stream after discarding its high 8 bits |
| *void writeChars(String s)* | **Write the String s as a sequence of characters** to the underlying output stream. |
| *void flush()* | **Sends the buffered bytes to the ouput stream** |

# DataInputStream /DataOutputStream Class

- Dataoutput1.java

## ObjectOutputStream Class

- It is a subclass of OutputStream.

- This class helps to **write objects to an output stream** as a series of bytes.

- **Object Written and Read should be in the same order.**

- Java provide mechanism to convernt the object to bytes. This process of conversion of an **object to a byte is called serialization.**

- The class whose object is to be sent to the ObjectOutputStream, **has to Serializable.**

# ObjectOutputStream Class

- ObjectOutputStream class has follow constructor:
  - ObjectOutputStream(OutputSteam ous);

# Methods of ObjectOutputStream Class

| Method | Purpose of the Method |
|---|---|
| *void writeObject(object ob)* | Writes the specified object to the ObjectOutputStream |
| *void write(int b)* | Write a single byte to the stream. |
| *void write(byte[] b)* | Write an array of bytes to the stream |
| *void write(byte[], int off, int len)* | Write a subrange of array of bytes of b, Starting at b[off], len bytes to the stream. |
| *void drain()* | Drains any buffered data to the stream |
| *void close()* | Close the stream |
| *void writeBoolean(boolean b)* | Write a boolean to the stream |

# Methods of ObjectOutputStream Class

| Method | Purpose of the Method |
|---|---|
| *void writeByte(int in)* | Writes a byte to the stream. |
| *void writeShort(int in)* | Write a 16 bits short to the stream. |
| *void writeChar(int ch)* | Write a 16 bits char to the stream. |
| *void writeInt(int in)* | Write a 32-bits int to the stream. |
| *void writeLong(long ln)* | Write a 64 bits long to the stream. |
| *void writeFloat(float fl)* | Write a 32 bits float to the stream. |
| *void writeDoble(double db)* | Write a 64 bits double to the stream. |

# Methods of ObjectOutputStream Class

| Method | Purpose of the Method |
| --- | --- |
| *void writeBytes(String s)* | Writes String s as a seuence of bytes to the stream. |
| *void writeChars(String s)* | Write String s as a sequence of chars |
| *void writeUTF(String s)* | Write the String s in UTF format |

**Programmes for the Objectoutput strems:**

- **Objectoutput1.java**
- **Objectoutput2.java**

# ObjectInputStream Class

- It is a subclass of InputStream.

- This class helps to **read objects from input stream** as a series of bytes.

- This stream, deserialization is done.

- That is, **the data are read from the ObjectInputStream and objects are reconstructed from bytes.**

- Only objects that supports the **Serializable interface can be read from the stream.**

# ObjectInputStream Class

- ObjectInputStream class has follow constructor:

    - ObjectInputStream(InputSteam ous);

# Methods of ObjectInputStream Class

| Method | Purpose of the Method |
|---|---|
| *object readObject()* | Read an object from the stream |
| *int read()* | Reads a byte of data, return -1 at the end of the stream |
| *int readByte(byte[] b, int off, int len)* | Read bytes from the stream into byte array starting at b[off] len bytes, returns the actual number of bytes or -1 when the end of the stream is reached |
| *int available()* | Returns the number of bytes that van be read from the stream. |
| *void close()* | Close the input stream |
| *boolean readBoolean()* | Reads in a boolean |
| *byte readByte()* | Reads an 8 bit byte |

# Methods of ObjectInputStream Class

| Method | Purpose of the Method |
|---|---|
| *int readUnsignByte()* | Reads an unsigned 8 bit byte |
| *short readShort()* | Reads a 16 bits short |
| *int readUnsignedShort()* | Read an unsigned 16 bit short |
| *char readChar()* | Reads a 16 bits char |
| *int readInt()* | Read a 32 bit int |
| *float readFloat()* | Reads a 32 bit float |
| *double readDouble* | Reads a 64 bit double |

# Methods of ObjectInputStream Class

| Method | Purpose of the Method |
|---|---|
| *void readFully(byte[] b)* | Reads bytes into b untill all bytes are read |
| *void readFully(byte[] b, int off, int len)* | Reads bytes into b, starting at b[off], len number of bytes |
| *int skipBytes(int len)* | Skips len numbers of bytes, returns the actual number of bytes skipped |
| *String  readUTF()* | Reads a UTF format String |

**Programmes for the Objectinput strems:**

- **Objectinput1.java**
- **Objectinput2.java**

# SequenceInputStream Class

- When there are **several input streams** from which data is to **be read, the SequenceInputStream class** is used.

- SequenceInputStream class has follow constructor:

  - SequenceInputStream(InputStream inps1, InputStream inps2)

  - SequenceInputStream(Enumeration Enumstream)

**Sequenceinput.java**

# Random Access Files

- Disk oriented random access files are handle in the **RandomAccessFile class.**

- **This stream is an isolated file stream. It as no superclass except for the super Object.**

- It also called **instant access files** or **direct access files**.

- When any file with large number of data, in particular random access file is vary useful.

- **It implements the DataInput and DataOutput interface.**

- Hence, methods in **RandomAccessFile, DataInputStream and DataOutputStream are common**.

# Random Access Files

- RandomAccessFile class has follow constructor:
  - *RandomAccessFile(String file, String access)*
    - Value is "r" for read only
    - Value is "rw" for read and write
  - *RandomAccessFile(File fobnect, String access)*
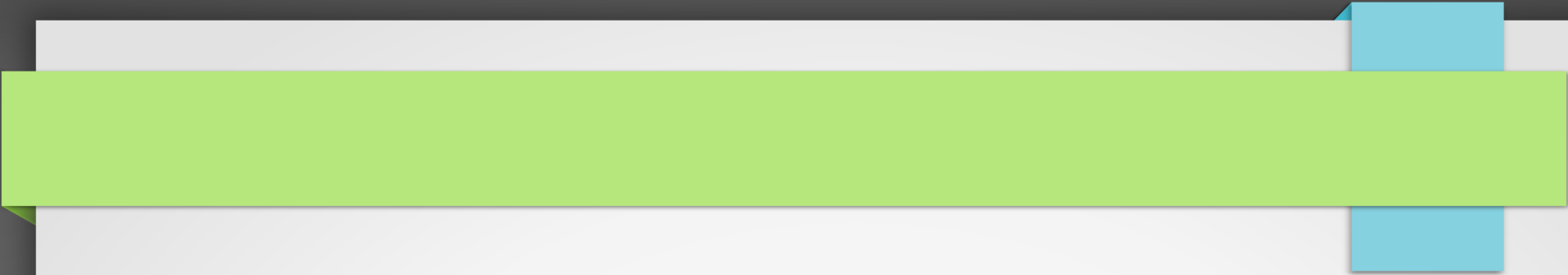
# Methods of RandomAccessFile Class

| Method | Purpose of the Method |
|---|---|
| *long getFilePointer()* | Returns the current location of the file pointer. |
| *void seek(long len)* | Sets the file pointer to len position from the beginning of the file. |
| *long length* | Returns the length of the file in bytes |
| *void setLength(long, newsize)* | Sets the length of this file to the size newsize |

**Programmes for the RandomAccessFile :**

- **Randomfilewrite.java**
- **Randomfileread.java**

# UNIT 1  COMPLETED

- Assignment Submission Date
  - Theory – 14 / 08 /2020
  - Practical –  / 7/ 2018
- CEC  Date
  - Theory -   07 / 08 / 2020
  - Practical –  06 / 08 / 2020