

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

```
graph TD; A[INTERNAL EVALUATION] --> B[ASSIGNMENTS]; A --> C[ATTENDANCE]; A --> D[MCQ TEST]; B --> E["5 Assignments * 5 Marks = 25 Marks"]; C --> F[25 Marks]; D --> G["20 Marks * 5 = 100 Marks"];
```

ASSIGNMENTS

**5 Assignments
* 5 Marks
= 25 Marks**

ATTENDANCE

25 Marks

MCQ TEST

**20 Marks * 5
= 100 Marks**

TEXT BOOK

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

Recommended Book

- Programming 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
- Filtered Byte Stream
- Sequence Input Stream
- Object Input / Output Stream
- Random Access File

UNIT -1 File Handling - “I/O 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**

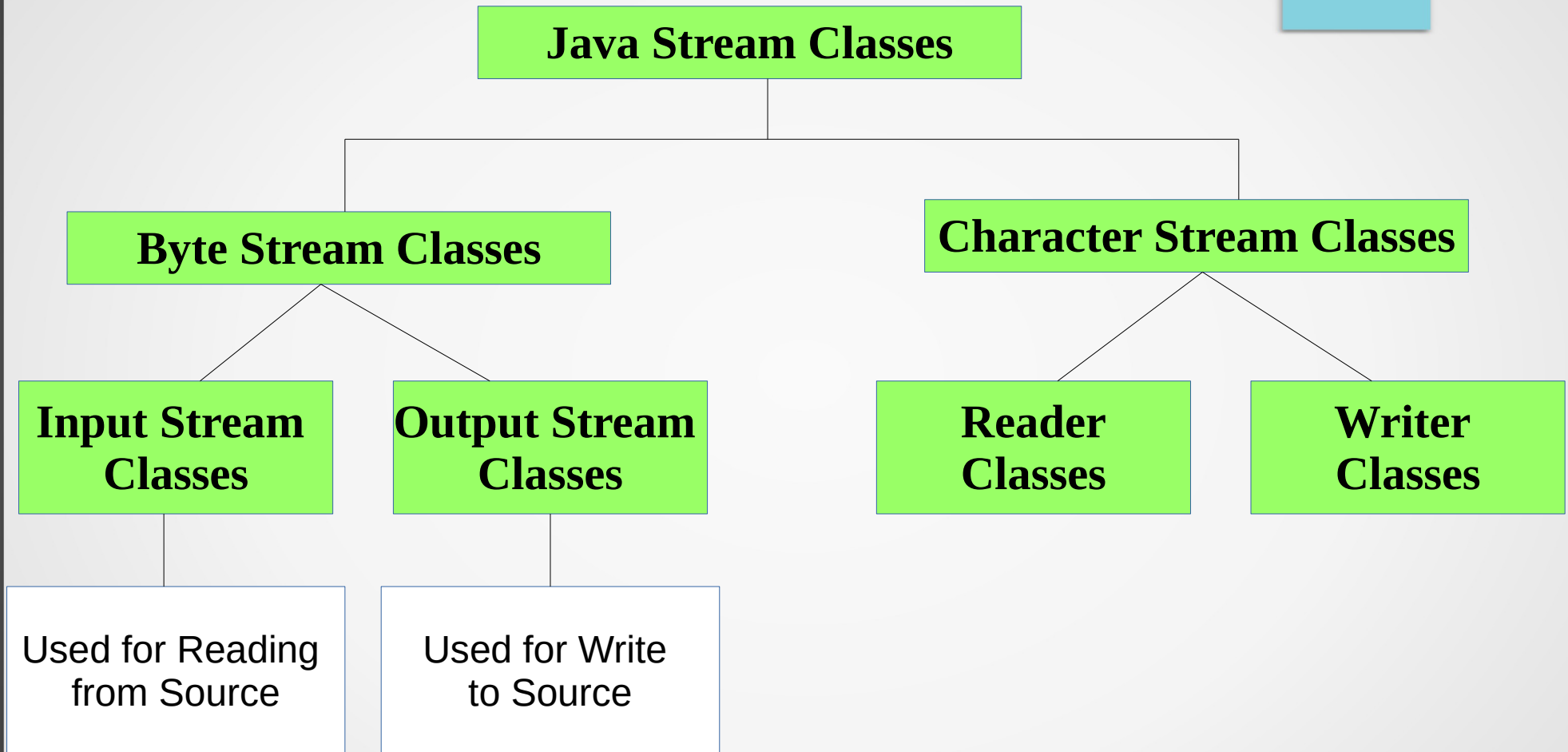
UNIT -1 File Handling - “I/O Stream”

- For Binary format I/O **Byte Streams** classes 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/O 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 classes or subclasses.
 - **Reader**
 - **Writer**

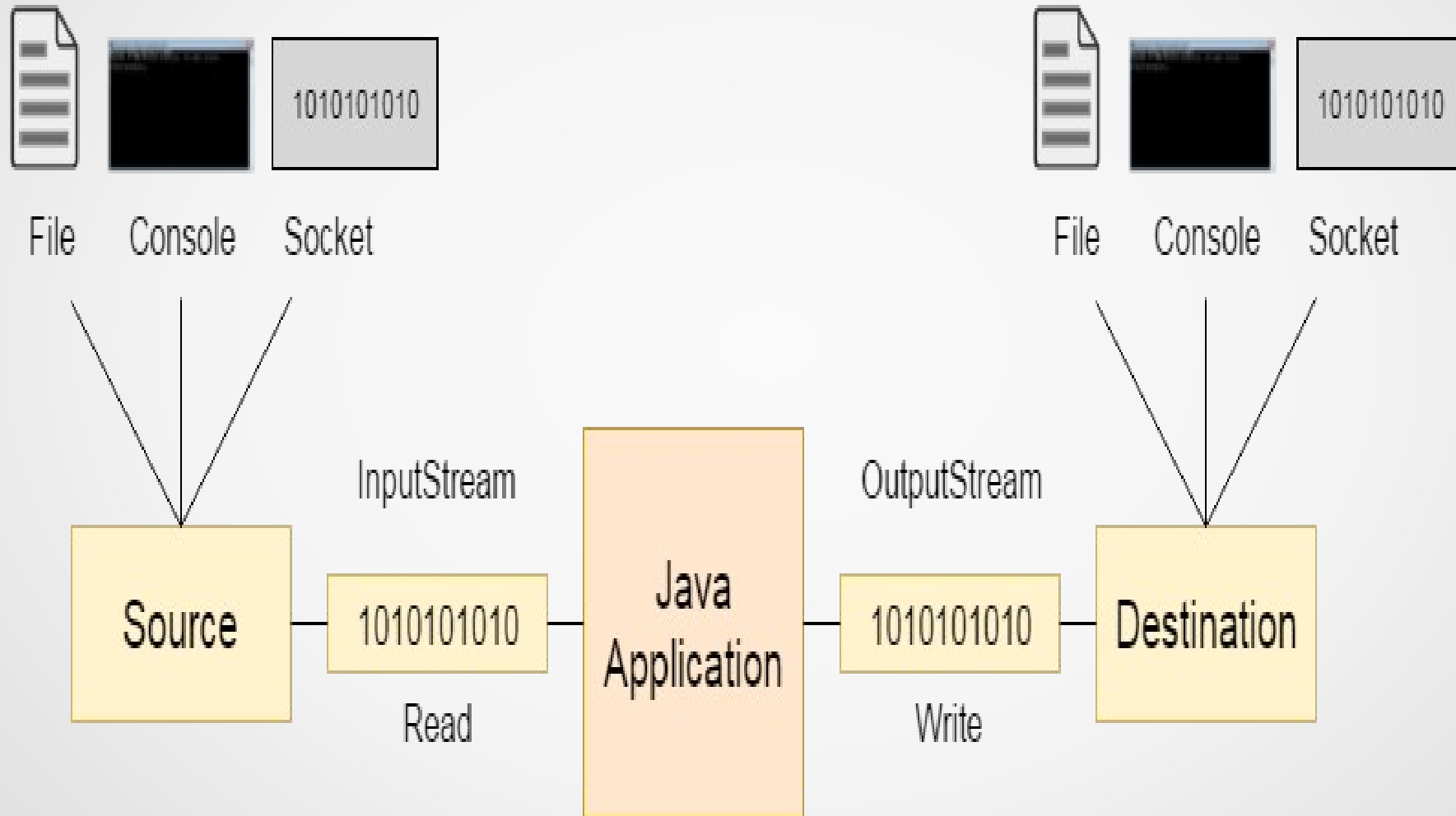
UNIT -1 File Handling - “I/O Stream”



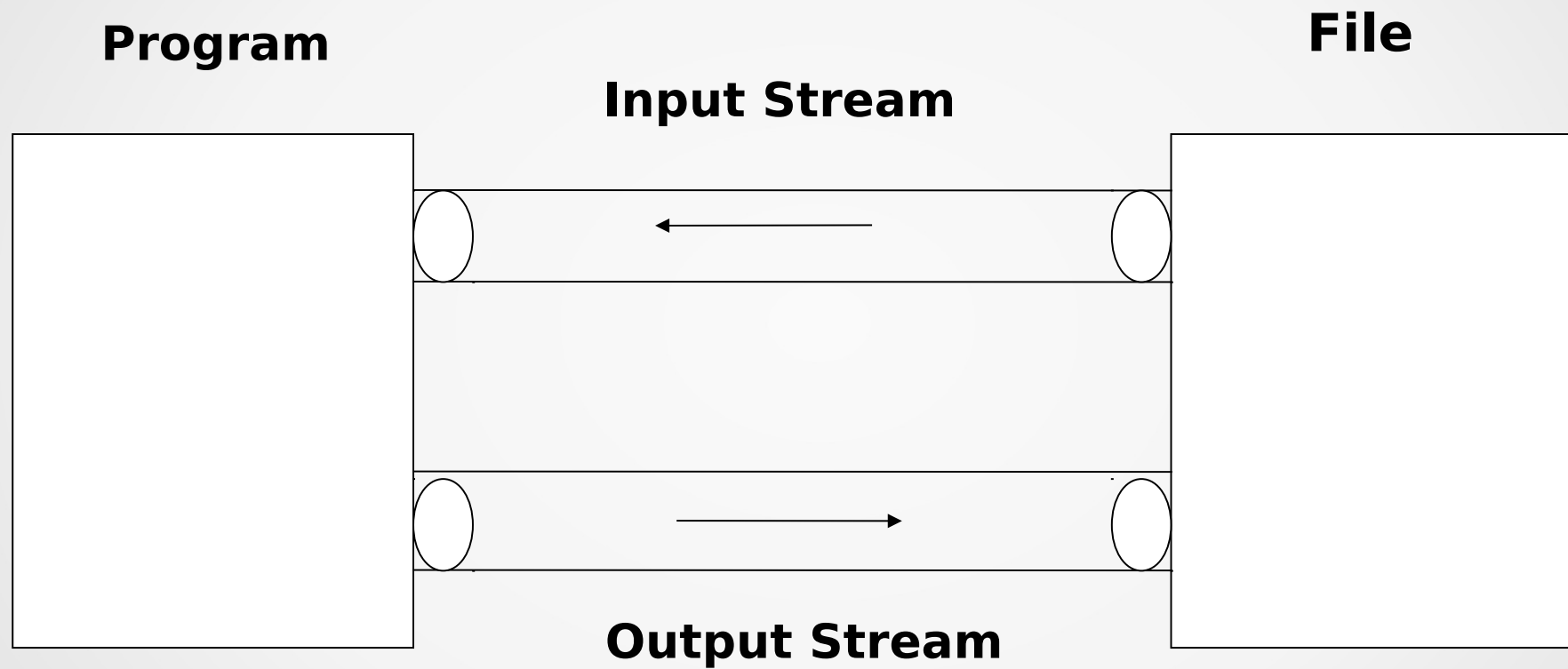
UNIT -1 File Handling - “I/O Stream”

- **Stream are two type:**
 - **Input Stream** – The sources from **data read**
 - **Output Stream** – The destination to **data write**

UNIT -1 File Handling - “I/O Stream”



UNIT -1 File Handling - “I/O 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
<i>String getName()</i>	Return the name of the directory or file.
<i>boolean exists()</i>	Return true if director or file exists else return false
<i>boolean canWrite()</i>	Return true if file object is writable else false.
<i>boolean canRead()</i>	Return true if file object is readable else false.
<i>boolean isDirectory()</i>	Return true if it is directory else false.

UNIT -1 File Class Methods

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

UNIT -1 File Class Methods

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

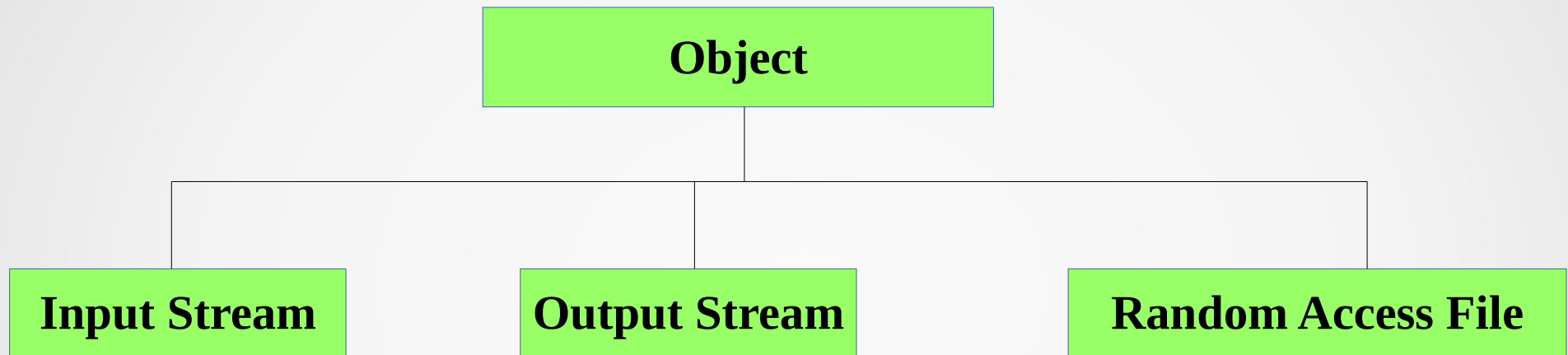
UNIT -1 File Class

- FileDemo.java

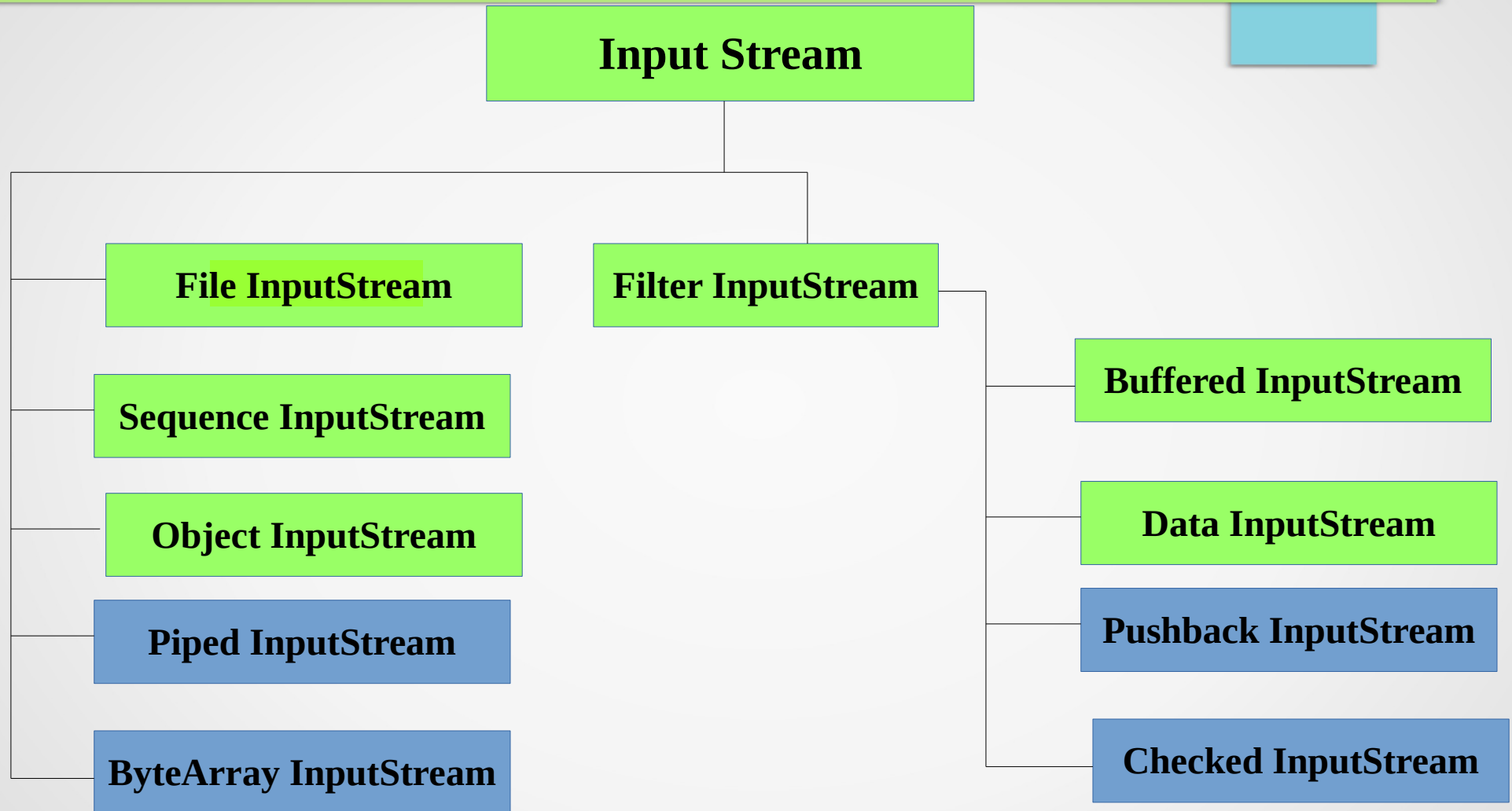
UNIT -1 I/O Stream

- 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**

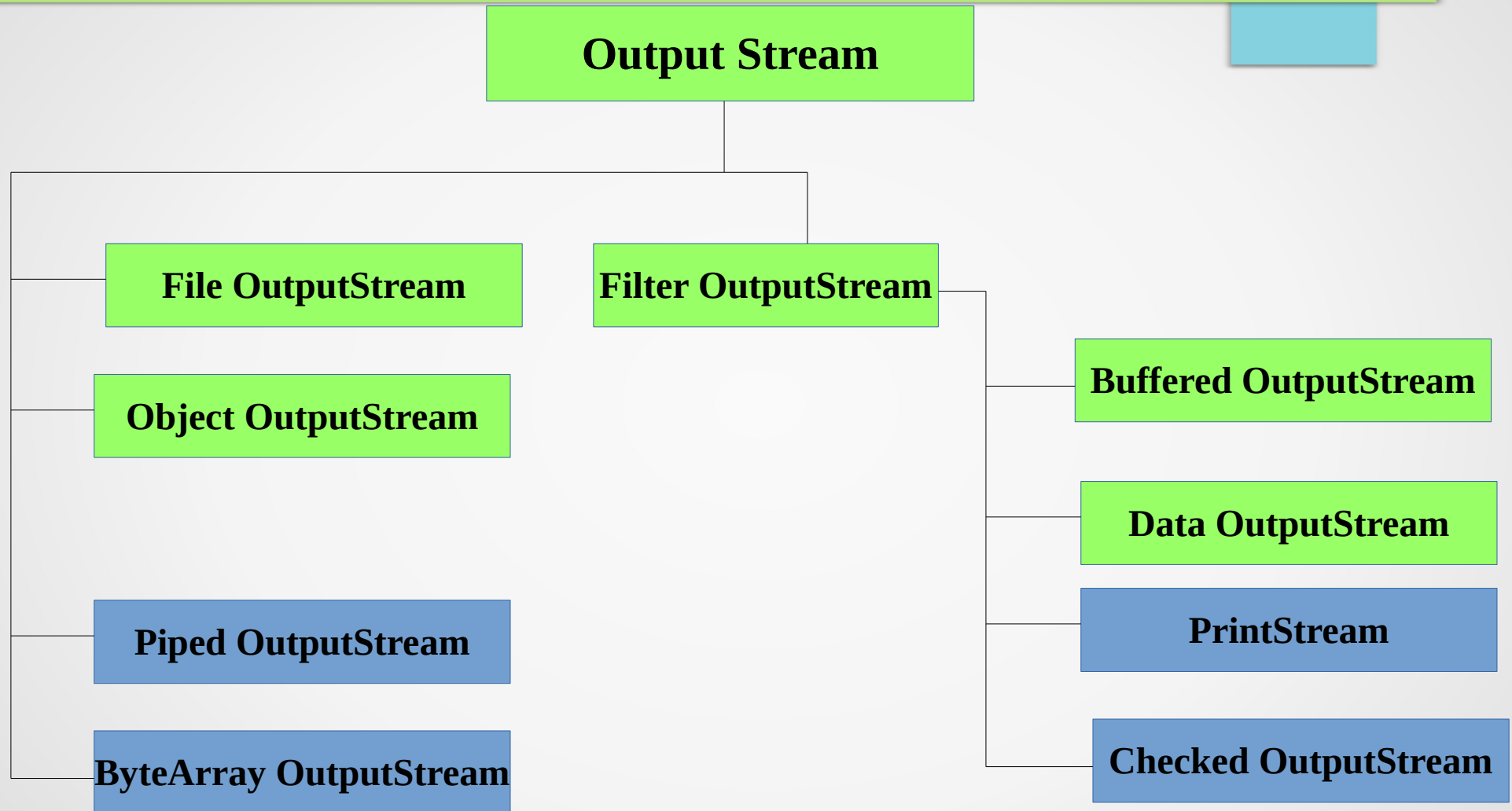
UNIT -1 I/O Stream



UNIT -1 I/O Stream



UNIT -1 I/O Stream



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 death 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
<i>abstract int read()</i>	Reads one byte and returns an int representation of that byte; returns -1 at the end of the system
<i>int read (byte ba[])</i>	Reads into an array of bytes 'ba' and returns the number of b.length bytes returns -1 at the end of the stream
<i>int read (byte ba[], int end, int length)</i>	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
<i>void close()</i>	Closes the input stream
<i>void mark(int marklen)</i>	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
<i>boolean marksupported()</i>	Returns true if the stream supports mark ()
<i>void reset ()</i>	The read control is reset to the previous set mark.
<i>int available ()</i>	Return the number of bytes available for reading
<i>long skip(long m)</i>	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
<i>abstract void write(int b)</i>	Writes a single byte b to an output stream
<i>void write (byte b[])</i>	Writes a array of byte b to an output stream
<i>void write (byte b[], int off, int len)</i>	Writes len number of bytes starting from b[off] to an output stream
<i>void close ()</i>	Closes the output stream
<i>void flush()</i>	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`

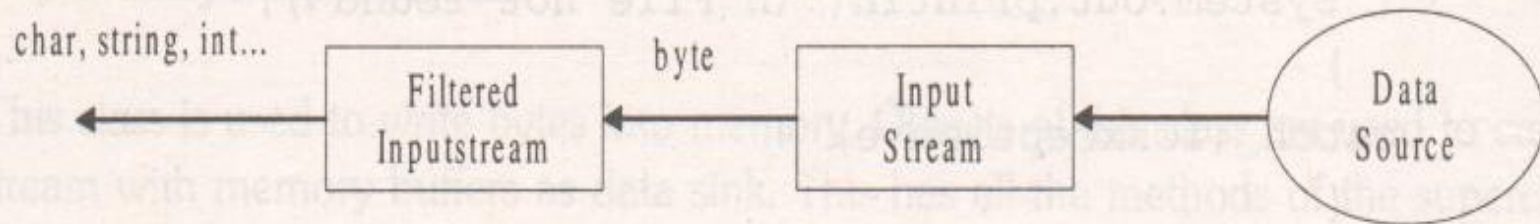
Practics

- 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.

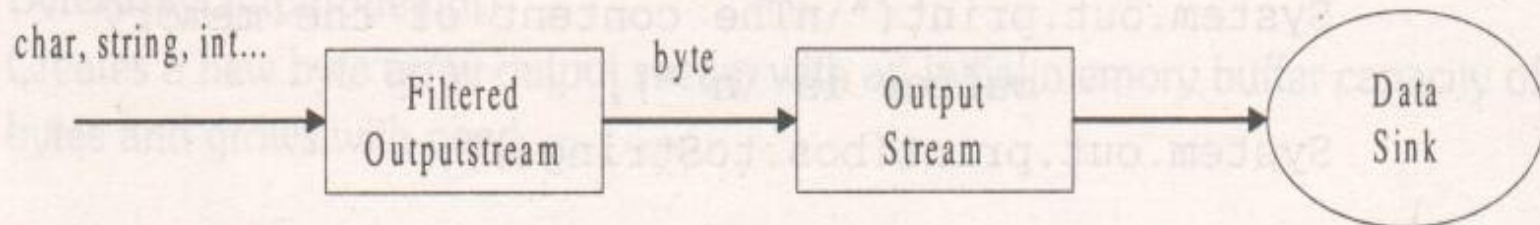
UNIT -1 Filtered Byte Stream

- The basic **byte streams** access the data in **byte form**.
- For **converting bytes to useful forms** such as char, string, int etc. **Java** have 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
<i>int read (byte[] b)</i>	Read bytes into the byte array b returns the numbers of bytes read; returns -1 at the end of the stream.
<i>int read (byte[] b, int off, int len)</i>	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.
<i>void readFully(byte[] b)</i>	Reads bytes into the byte array b, the maximum bytes read being the capacity of the byte array .
<i>Boolean readBoolean()</i>	Reads a boolean value from the input stream
<i>Byte readByte()</i>	Reads an 8bit signed byte from the input stream

Methods of DataInputStream Class

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

Methods of DataInputStream Class

Method	Purpose
<i>int skipBytes(int n)</i>	Skips n bytes and returns the actual number of bytes skipped.
<i>int readInt()</i>	Read 4 bytes from the input stream and treats it as an int.
<i>double readDouble()</i>	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 underlying output stream.
- Constructure :
 - `DataStream(OutputStream ous)`

Methods of DataOutputStream Class

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

Methods of DataOutputStream Class

Method	Purpose
<i>void writeChar(int in)</i>	Write a char as 2 byte value to the underlying stream.
<i>void writeInt(int in)</i>	Write an int as 4 bytes to the underlying stream.
<i>void writeLong(long ln)</i>	Write a long as 8 bytes to the underlying stream.
<i>void writeFloat(float f)</i>	Converts the float to an int using a special method and writes the int as 4 bytes to the underlying stream.
<i>void writeDouble(double d)</i>	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
<i>void writeBytes(String s)</i>	Write the String s as a sequence of bytes to the underlying output stream after discarding its high 8 bits
<i>void writeChars(String s)</i>	Write the String s as a sequence of characters to the underlying output stream.
<i>void flush()</i>	Sends the buffered bytes to the output 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(OutputStream ous);`

Methods of ObjectOutputStream Class

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

Methods of ObjectOutputStream Class

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

Methods of ObjectOutputStream Class

Method	Purpose of the Method
<i>void writeBytes(String s)</i>	Writes String s as a sequence of bytes to the stream.
<i>void writeChars(String s)</i>	Write String s as a sequence of chars
<i>void writeUTF(String s)</i>	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
<i>object readObject()</i>	Read an object from the stream
<i>int read()</i>	Reads a byte of data, return -1 at the end of the stream
<i>int readByte(byte[] b, int off, int len)</i>	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
<i>int available()</i>	Returns the number of bytes that can be read from the stream.
<i>void close()</i>	Close the input stream
<i>boolean readBoolean()</i>	Reads in a boolean
<i>byte readByte()</i>	Reads an 8 bit byte

Methods of ObjectInputStream Class

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

Methods of ObjectInputStream Class

Method	Purpose of the Method
<i>void readFully(byte[] b)</i>	Reads bytes into b untill all bytes are read
<i>void readFully(byte[] b, int off, int len)</i>	Reads bytes into b, starting at b[off], len number of bytes
<i>int skipBytes(int len)</i>	Skips len numbers of bytes, returns the actual number of bytes skipped
<i>String readUTF()</i>	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

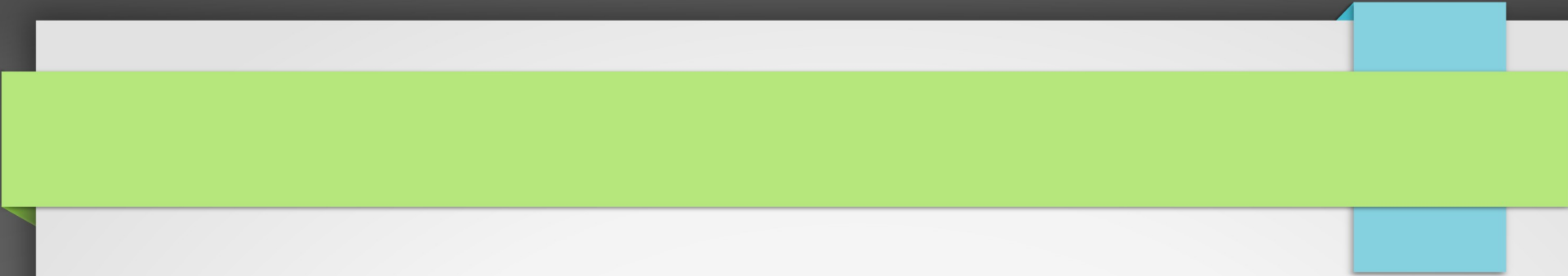
Method	Purpose of the Method
<i>long getFilePointer()</i>	Returns the current location of the file pointer.
<i>void seek(long len)</i>	Sets the file pointer to len position from the beginning of the file.
<i>long length</i>	Returns the length of the file in bytes
<i>void setLength(long, newsize)</i>	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