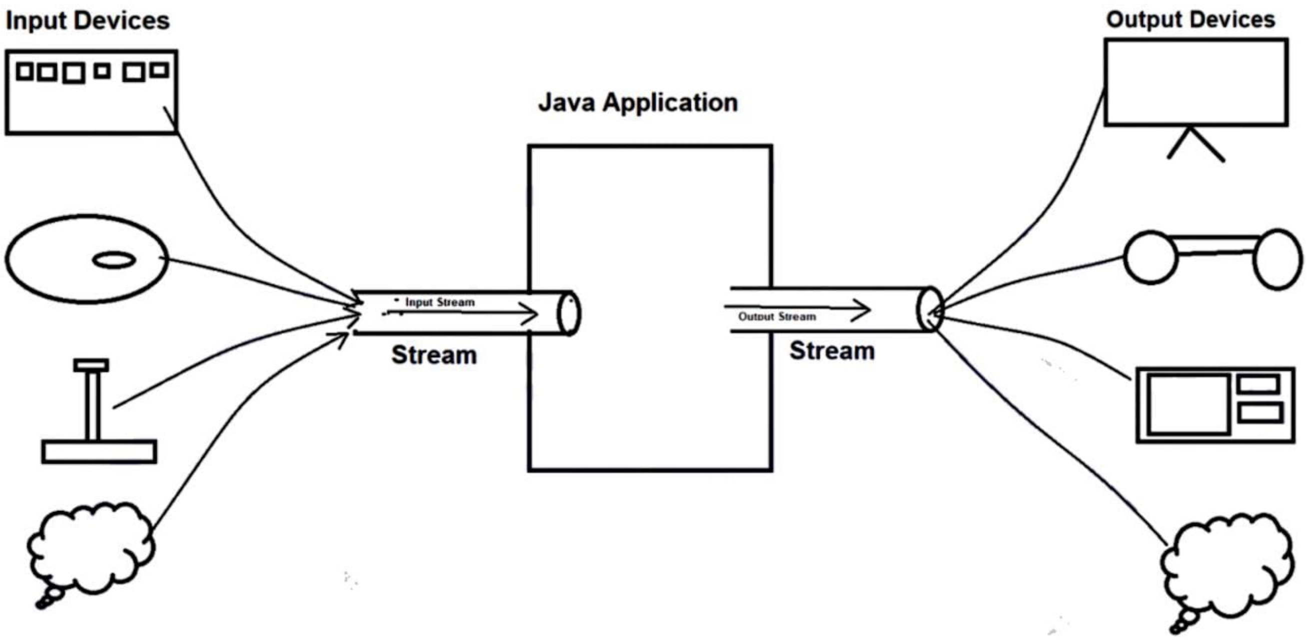
Java IO :-

Terms used in Java IO :-

1. Input Devices : Keyboard, mouse, Optical pen, Joystick etc
2. Output Devices : Screen, Head Set, Printers, Speakers etc
3. Stream : Flow of data, Real world example Water flow

=> **Stream** : Stream is a medium which allow the data to flow from input devices to java application or from java application to output devices



-> In java there are 3 predefined stream which transfer the data from CMD to java application which are as follows :-

1. System.out
2. System.in
3. System.err

-> To transfer the data from input devices to java application or java application to output devices, java has provided many classes which are present in java.io package

-> Types of Streams :-

1. Byte Stream
2. Character Stream

=> Byte Stream :-

-> In this stream data is transfered in the form of bytes i.e. 0 and 1

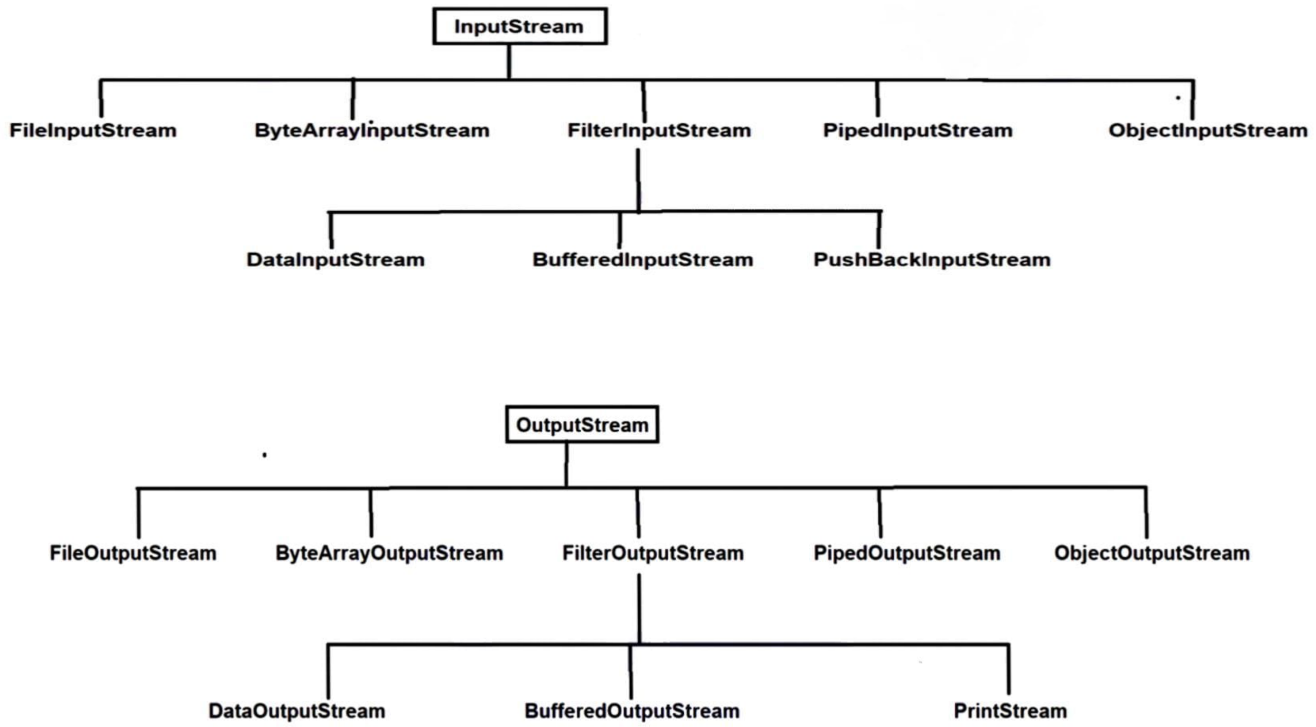
-> The length of byte stream is 1 byte

-> Types of byte stream :-(Both are abstract class)

1. InputStream
    - -> public int read(-) throws IOException
    - -> public int available() throws IOException
    - -> public void close() throws IOException

2. OutputStream
    - -> public void write(-) throws IOException
    - -> public void flush() throws IOException
    - -> public void close() throws IOException

-> Hierarchy of InputStream and OutputStream classes :-



* ‘out’ is a static variable of PrtingStream Class

==

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;

public class FIS {
    public static void main(String[] args) throws Exception {
        FileInputStream fis = new FileInputStream("src/ScannerDemo/Scanner1.java") ;
        int i = fis.read() ;
        System.out.println((char)i);
    }
}
```

===

```
public class FIS {
    public static void main(String[] args) throws Exception {
        FileInputStream fis = new FileInputStream("src/ScannerDemo/Scanner1.java") ;
        int i ;
        while((i= fis.read())!=-1){
            System.out.print((char)i);
        }
        fis.close();
    }
}
```

===

```
public class ReadData1
{
    public static void main(String[] args)
    {
        FileInputStream fis=null;
        try
        {
            fis=new FileInputStream("src/ScannerDemo/Scanner1.java");
            int i;
            while( (i=fis.read()) != -1 )
            {
                System.out.print((char)i);
            }
        }
        catch(IOException e)
        {
            System.out.println(e);
        }
        finally
        {
            try
            {
                fis.close();
            }
            catch(IOException e)
            {
                System.out.println(e);
            }
        }
    }
}
```

===

try with resource-> no need to close the resource now

```
public class ReadData2
{
    public static void main(String[] args)
    {
        try(
            FileInputStream fis=new FileInputStream("E:\\javabatch\\PackageDemo.java");
         )
        {
            int i;
            while( (i=fis.read()) != -1 )
            {
                System.out.print((char)i);
            }
        }
        catch(IOException e)
        {
            System.out.println(e);
        }
    }
}
```

Write Demo:

```
public class WriteData
{
    public static void main(String[] args)
    {
        try
        {
            String data="Arun Kumar Sharma";

            FileOutputStream fos=new FileOutputStream("/Users/arunkumarsharma Desktop/javaio.txt");

            byte[] b=data.getBytes();
            fos.write(b);
            fos.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
```

* Automatically create the file

=> FileOutputStream fos=new FileOutputStream("/Users/arunkumarsharma/Desktop/javaio.txt",true);

true-> to append in the prevoius file

=>FileOutputStream fos=new FileOutputStream("/Users/arunkumarsharma/Desktop/javaio.txt”,false);

false-> to replace the text in the prevoius file

Read from 1 file and write it in Another File

```
public class WriteData {
    public static void main(String[] args) {
        FileInputStream fis = null ;
        FileOutputStream fos = null ;
        {
            try {
                fis = new FileInputStream("/Users/arunkumarsharma/Documents/FileIO/Demo.java");
                fos = new FileOutputStream("/Users/arunkumarsharma/Documents/FileIO/Sum.txt");
                int i;
                while ((i = fis.read()) != -1) {
                    fos.write(i);
                }
            }
            catch (Exception e) {
                System.out.println(e);
            }
            finally {
                try {
                    fis.close();
                    fos.close();
                }
                catch (Exception e) {
                    System.out.println(e);
                }
            }
        }
    }
}
```

copying image:

```
public class WriteData {
    public static void main(String[] args) {
        FileInputStream fis = null ;
        FileOutputStream fos = null ;
        {
            try {
                fis = new FileInputStream("/Users/arunkumarsharma/Documents/FileIO/Lion.jpg");
                fos = new FileOutputStream("/Users/arunkumarsharma/Documents/FileIO/newLion.jpg");
                int i;
                while ((i = fis.read()) != -1) {
                    fos.write(i);
                }
            }
            catch (Exception e) {
                System.out.println(e);
            }
            finally {
                try {
                    fis.close();
                    fos.close();
                }
                catch (Exception e) {
                    System.out.println(e);
                }
            }
        }
    }
}
```

=> Character Stream :

-> It is used to transfer the data in characters form

-> In character stream the length of data is 2 bytes

-> Types of Character Stream :-

1. Reader
    - -> read(-)
    - -> close()
2. Writer
    - -> write()
    - -> close()

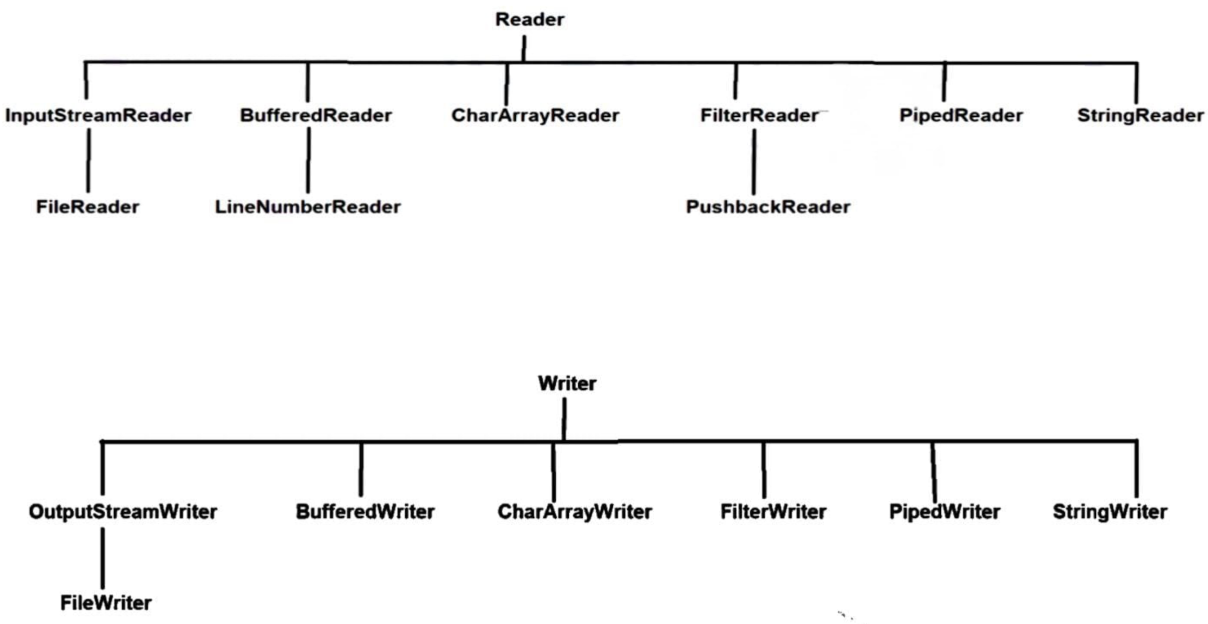
=> Dynamic Input Approches :-

-> Dynamic Input Approch is the way by which programmer take the input value at runtime

-> There are 3 ways for dynamic input approch

1. Scanner
2. BufferedReader
3. Console

-> Hierarchy of Writer & Reader classes



Read:

```
public class ReaderDemo {
    public static void main(String[] args) {
        try{
            FileReader fr = new FileReader("src/StudentDemo/ThisDemo1.java") ;
            int i ;
            while((i=fr.read())!=-1){
                System.out.print((char)i);
            }
            fr.close();
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }
}
```

==

Write:

```
public class ReaderDemo {
    public static void main(String[] args) {
        String str = "This is my first Character Stream Demo" ;
        try{
            FileWriter fw = new FileWriter("/Users/arunkumarsharma/Desktop/charracterstream1.txt") ;
            fw.write(str);   //no need of conversion this time, earlier we need to change it into byte array
            fw.close();
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }
```

Scanner :

```
public class ScannerDemo
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter Email : ");
        String email1=s.next();
        System.out.println("Enter Password : ");
        String pass1=s.next();
        if(email1.equals("admin@gmail.com") && pass1.equals("admin123"))
        {
            System.out.println("login successfully");
        }
        else
        {
            System.out.println("login failed");
        }
    }
}
```

BufferedReader:-> only int and String input

```
public class BufferedReaderDemo
{
    public static void main(String[] args)
    {
        try
        {
           // InputStreamReader isr=new InputStreamReader(System.in);
           // BufferedReader br=new BufferedReader(isr);

            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Enter Name : ");
            //int i=br.read();
            String s=br.readLine();
            System.out.println(s);

            br.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

Console:

```
public class ConsoleDemo {
    public static void main(String[] args) {
        Console c = System.console() ;

        System.out.println("Enter Email..");
        String email = c.readLine() ;

        System.out.println("Enter Password..");
        char[] pass = c.readPassword() ;
        String password = new String(pass) ;

        if(email.equals("arun@123") && password.equals("admin123")){
            System.out.println("Login Successfull");
        }
        else{
            System.out.println("Failed Login");
        }
    }
```