**Teacher’s Name:- Ms. Sonia Yogi**

**Assistant Professor**

**Department of Physics and Electronics**

**Course Name:- B.Sc.(H)Electronics (2nd Year)**

**Paper Name:- Internet and Java Programming (SEC)**

**Unique Paper Code:- 32513907**

**Study material for 16/03/2020 – 21/03/2020**

***File Handling***

* What is file handling in java?
* What is a stream?
* JAVA FILE METHODS
* File operations in java

File handling implies how to read from and write to file in java. Java provides the basic I/O package for reading and writing streams. Java.io package allows to do all input And output task in java.

In order to use file class, you need to create a object of the class and specify name or directory name.

First you will write import java.io.file that is used to import the file class and then you have to create object of the file and specify the file name and location.

JAVA used the concept of stream to make input and output operations in file.

***Stream : Sequence of data***

Two types of stream

1. Byte stream (it mainly incorporates with byte data i.e. when i/p and o/p happen with byte data then it is called the file handling with byte stream)
2. Character stream

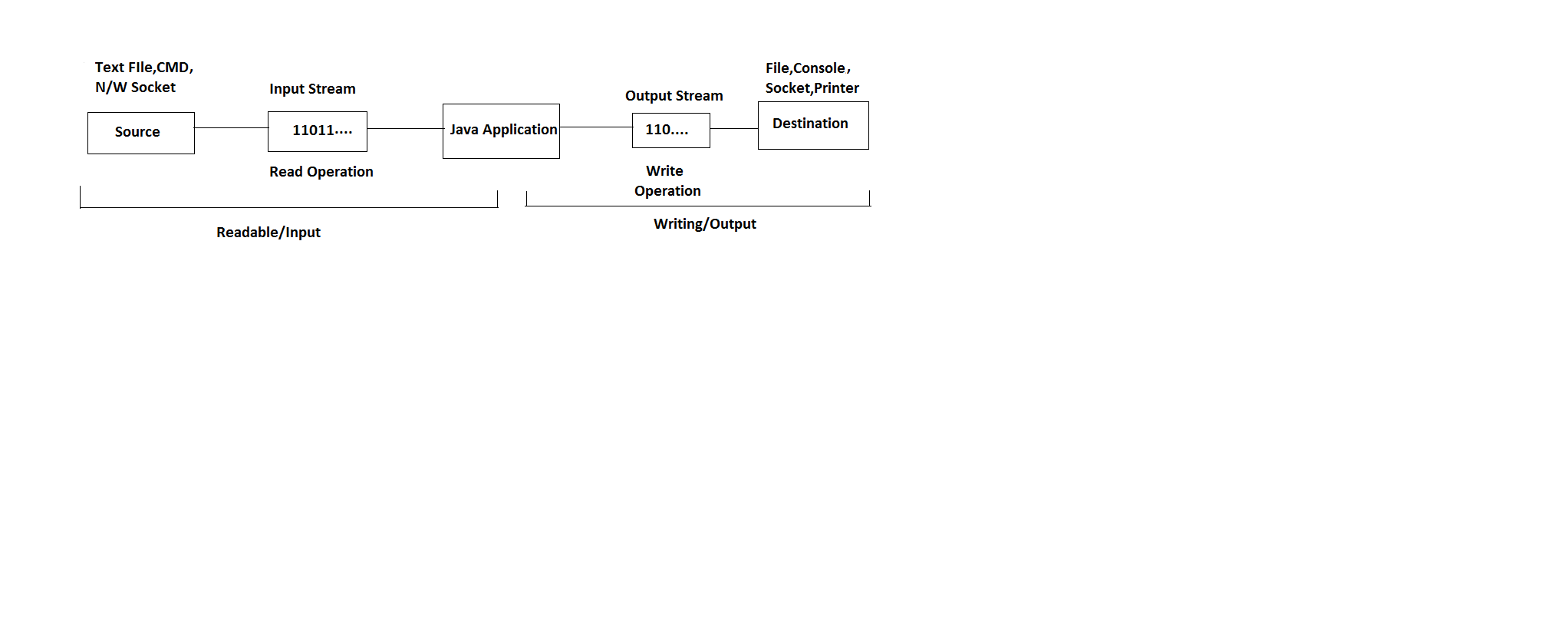
* Incorporate with character
* When i/p and o/p process happen with character then it is called file handling with character stream.

**File methods**

1. canRead() ------------------ Test whether file is readable or not
2. canWrite() ------------------ Test whether file is writable or not
3. createNewFile() ------------create an empty file
4. delete() ------------------------delets a file
5. exists()------------------------test whether the file exists or not
6. getname()--------------------returns the name of file
7. getAbsolutePath() ---------returns the path of the file
8. length()-----------------------returns the size of the file in byts
9. list() --------------------------returns the array of the file in directory
10. mkdir()-----------------------create a directory

***Stream – sequence of data composed of bytes***

* System.out
* System.in
* System.err



O/P stream class

* Void write(int) - throws ioexception (write bytes)
* Void write(bytes[]) - throws ioexception (array of bytes)
* Void flush() - throws ioexception (flushes current stream)
* Void close() - throws ioexception (close current stream)

*I/P stream class*

* Int read() - throws ioexception (next bytes)
* Int available() - throws ioexception (returns number of available bytes)
* Void close() - throws ioexception (close current i/p stream)

NOTE:- **All the methods of input & output stream must be lie in try catch block.**

These methods are used to preform various file operations in java

*File Operations in Java*

1. Create a file
2. Get file information
3. Write to a file
4. Read a file

In order to create a file, you can use “createNewFile()” method this method returns true if file is successfully created and returns false if it fails or the file already exists.

Program 1:-**Program to create a File using File Handling.**

package Filehandling;

import java.io.IOException;

import java.io.File;

public class Files {

public static void main(String args[])

{

try

{

File obj=new File("D:\\filehandle\\file1.txt");

if(obj.createNewFile())

{

System.out.println("File is created:"+obj.getName());

}

else

{

System.out.println("File already exists.");

}

}

catch(IOException e)

{

System.out.println("An error occurred.");

}

}

}

**Output:-**

1. **When code runs first time:-**

File is created:file1.txt

1. **When code runs Second time:-**

File already exists.

Program 2:- **Program to get a File Information using FileHandling.**

package Filehandling;

import java.io.IOException;

import java.io.File;

public class Fileinfo {

public static void main(String args[])

{

File obj=new File("D:\\filehandle\\file1.txt");

if(obj.exists())

{

System.out.println("File Name:-"+obj.getName());

System.out.println("Path:-"+obj.getAbsolutePath());

System.out.println("File size in bytes:-"+obj.length());

System.out.println("Writable:-"+obj.canWrite());

System.out.println("Readable:-"+obj.canRead());

}

else

{

System.out.println("File does not exist.");

}

}

}

**Output:-**

1. **Output when Program 3 & 4 is not made:-**

File Name:-file1.txt

Path:-D:\filehandle\file1.txt

File size in bytes:-0

Writable:-true

Readable:-true

1. **Output when Program 3 & 4 is made:-**

File Name:-file1.txt

Path:-D:\filehandle\file1.txt

File size in bytes:-31

Writable:-true

Readable:-true

Program 3:- **Program to write in a File using FileHandling.**

package writefile;

import java.io.IOException;

import java.io.File;

import java.io.FileWriter;

public class writefile {

public static void main(String args[])

{

try

{

FileWriter obj=new FileWriter("D:\\filehandle\\file1.txt");

obj.write("Java is the Prominent Language.");

obj.close();

System.out.print("Successfully wrote in the File.");

}

catch(IOException e)

{

System.out.print("An error occurred.");

e.printStackTrace();

}

}

}

**Output:-**

Successfully wrote in the File.

Program 4:- **Program to read a File using FileHandling.**

package Filehandling;

import java.util.Scanner;

import java.io.IOException;

import java.io.File;

import java.io.FileNotFoundException;

public class Readfile {

public static void main(String[] args)

{

try

{

File obj=new File("D:\\filehandle\\file1.txt");

Scanner reader=new Scanner(obj);

while(reader.hasNextLine())

{

String data=reader.nextLine();

System.out.println(data);

}

reader.close();

}

catch(FileNotFoundException e)

{

System.out.println("An error occurred.");

e.printStackTrace();

}

}

}

**Output:-**

Java is the Prominent Language.