Reading and Writing Files

**Write File-**

Step 1: Use an output stream which selects the directory where file will be save.

Step 2: Use a same output stream to create a file and chain it to previous step.

Step 3: BufferedWriter class to write characters in file and chain it to previous step.

Step 4: Close connection (last stream) imp: because close the resources to release.

Example:

package readingAndWriting;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

public class WriteAFile {

public static void main(String[] args) throws IOException {

//stream connectivity

File f = new File("D:\\FileHandlingBatch56\\text1.html");

FileWriter fw = new FileWriter(f);

BufferedWriter writer = new BufferedWriter(fw);

//writing code

writer.write("Second Line, Mid");

writer.newLine();

writer.write("Third Line, End");

writer.newLine();\*/

//close stream

writer.close();

System.out.println("File is created....");

}

}

---------------------------------------------------------------------------------------------

**Read File-**

Step 1: Use an input stream which selects the directory in which our file is saved.

Step 2: Use a same input stream to use a file and chain it to previous step.

Step 3: Use BufferedReader class to read characters in file and chain it to previous step.

Step 4: Close connection (last stream) imp: because close the resources to release.

Example:

package readingAndWriting;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

public class ReadAFile {

public static void main(String[] args) throws IOException {

//stream connectivity

File f = new File("D:\\FileHandlingBatch56\\text1.txt");

FileReader fr = new FileReader(f);

BufferedReader reader = new BufferedReader(fr);

//Read a file

String line = null;

while((line=reader.readLine())!=null){

System.out.println(line);

}

//close

reader.close();

}

}