**LAB 7.1**

*Write a program, wheresystem will read data form one text file and will write it on other text file using byte stream.*

**Steps:**

* Write a class IOPractice (Having p.s.v.main method)
* Write a method which reads data from inFile.txt and write it on outFile.txt

**package** tdi.javaFLP.languageBasic;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**public** **class** IOPractice {

/\*\*

\* **@throws** IOException

\*/

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

IOPractice ioPractice = **new** IOPractice();

ioPractice.byteStreamPrac();

}

/\*\*

\* For Byte stream, we use FileInputStream and FileOutputStream

\* **@throws** IOException

\*/

**private** **void** byteStreamPrac()**throws** IOException {

FileInputStream in = **null**;

FileOutputStream out = **null**;

**try** {

in = **new** FileInputStream("d:/inFile.txt");

out = **new** FileOutputStream("d:/outFile.txt");

**int** c;

**while**((c = in.read()) != -1){

out.write(c);

}

} **catch** (FileNotFoundException e) {

e.printStackTrace();

} **finally** {

**if** (in != **null**){

in.close();

}

**if** (out != **null**){

out.close();

}

}

}

}

**LAB 7.2**

*Do the task of lab 7.1, but this time use character stream*

**Steps:**

* Write a class IOPractice (Having p.s.v.main method)
* Write a method which reads data from inFile.txt and write it on outFile.txt

**package** tdi.javaFLP.languageBasic;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**public** **class** IOPractice {

/\*\*

\* **@throws** IOException

\*/

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

IOPractice ioPractice = **new** IOPractice();

// ioPractice.byteStreamPrac();

ioPractice.charStreamPrac();

}

/\*\*

\* For character stream, we use FileReader and FileWriter

\* All everything is same as byte stream

\* Character stream uses byte stream for actual physical I/O

\* **@throws** IOException

\*/

**private** **void** charStreamPrac()**throws** IOException {

FileReader in = **null**;

FileWriter out = **null**;

**try** {

in = **new** FileReader("d:/inFile.txt");

out = **new** FileWriter("d:/outFile.txt");

**int** c;

**while**((c = in.read()) != -1){

out.write(c);

}

} **catch** (FileNotFoundException e) {

e.printStackTrace();

} **finally** {

**if** (in != **null**){

in.close();

}

**if** (out != **null**){

out.close();

}

}

}

}

**Learning:**

* All byte stream classes are descended from [InputStream](http://download.oracle.com/javase/7/docs/api/java/io/InputStream.html) and [OutputStream](http://download.oracle.com/javase/7/docs/api/java/io/OutputStream.html)
* The most important difference is that Characters stream uses FileReader and FileWriter for input and output in place of FileInputStream and FileOutputStream
* In Characters stream, the int variable holds a character value in its last 16 bits; while in Bytes stream, the int variable holds a byte value in its last 8 bits

**LAB 7.3**

*Convert FileReader and FileWriter into BufferReader and BufferWriter*

**Steps:**

* Modify the code of Lab 3.2 – change the unbuffered code into buffered code

**Unbuffered (previous code)**

FileReader inputStream = new FileReader("xanadu.txt");

FileWriter outputStream = new FileWriter("characteroutput.txt");

**Buffered (change the above code as below)**

inputStream = new BufferedReader(new FileReader("xanadu.txt"));

outputStream = new BufferedWriter(new FileWriter("characteroutput.txt"));

**Learning:**

* How to use buffered stream
* Benefit of buffered stream

**LAB 7.4**

*Use BufferReader (create it with the help of InputStreamReader) to read data from keyboard and display (write) it on console*

**package** tdi.javaFLP.languageBasic;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**public** **class** IOPractice {

/\*\*

\* **@param** args

\* **@throws** IOException

\*/

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

IOPractice ioPractice = **new** IOPractice();

ioPractice.standardReaderPractice();

}

**private** **void** standardReaderPractice() **throws** IOException{

InputStreamReader streamReader = **new** InputStreamReader(System.*in*);

BufferedReader bufferedReader = **new** BufferedReader(streamReader);

System.*out*.println("Enter your name : ");

String str = bufferedReader.readLine();

System.*out*.println("Hi " + str);

}

}

**OUTPUT**

Enter your name :

vivek

Hi vivek

**Learning:**

* Use of Buffer reader to read the keyboard (user’s input at console)
* How to make an interactive program using Buffer reader