PRESIDENCY UNIVERSITY

(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)

CSE3146 – Advanced JAVA Programming

LAB SHEET - 2

Module 2 – Input Output Operation in Java

Operations with Text files

Q1. Write a Java program to perform the following operations with text files.

a) Create a folder , student name as folder name in c drive by passing the folder name at run time using Scanner.
b) Create a file to write an essay “shortnnoteson multithreading”, file name must be “multithread.txt”.
c) Read the content of the file to output console.
d) Copy the content of above file to another file studentname_multithread_copy.txt.

Solution :

a) Creating folder

```
import java.io.*;
import java.util.Scanner;
public class CreateFolder
{
public static void main(String arr[])
{
Scanner sc = new Scanner(System.in);
System.out.println("enter folder name");
String foldername=sc.next();
File f=new File(foldername);
if(f.exists()&&f.isDirectory())
System.out.println("already exist");
else if(f.mkdir()==true)
System.out.println("successfully created");
else
System.out.println("cannot be created");
}
}
```

b) Creating file

```
import java.io.FileWriter;
import java.util.Scanner;
public class Main {
public static void main(String args[]) throws Exception {
String essays = "This is the data in the output file\nThis is the data in the output file\n This is the data in the output file\n";
Scanner sc = new Scanner(System.in);
System.out.println("Enter file name and folder name");
String filename=sc.next();
String foldername=sc.next();
FileWriter output = new FileWriter(foldername+"\\"+filename+".txt");
System.out.println("file created");
output.write(essays);
System.out.println("file written");
output.close();
}}
```

c) Reading File

```
import java.io.BufferedReader;
Import java.io.File;
Import java.io.FileReader;
public class Main {
public static void main(String args[]) throws Exception {
File fileob=new File("linus\\multithread.txt");
FileReader fr = new FileReader(fileob);
BufferedReader br = new BufferedReader(fr);
String line;
while((line=br.readLine())!=null) {
System.out.println(line);
}
br.close();
}}
```

d) Copying File

```
//either PrintWriter or BufferWriter
import java.io.*;
public class Main {
public static void main(String args[]) throws Exception {
File source=new File("linus\\multithread.txt");
File target=new File("linus\\multithread_copy.txt");
FileReader fr = new FileReader(source);
FileWriter fw = new FileWriter(target);
BufferedReader br = new BufferedReader(fr);
//BufferedWriter bw = new BufferedWriter(fw);
PrintWriter pw = new PrintWriter(fw, false);
String line;
while((line=br.readLine())!=null) {
pw.write(line);  //bw.write(line);
pw.println(); //bw.newLine();
}
System.out.println("1 file copied");
br.close();
//bw.close();
pw.close();
}
}
```

Q2. Write a Java program to perform the following operations with text files.

a) Display the number of characters, sentences and words present in a text file multithread.txt.

b) Display the content of the file on the screen with line number before each line.

Solution :

```
import java.io.*;
public class Test {
public static void main(String[] args) throws Exception {
File file = new File("linus\\multithread.txt");
FileReader fr = new FileReader(file);
BufferedReader br = new BufferedReader(fr);
String line;
int wordCount = 0;
int characterCount = 0;
int sentenceCount = 0;
int linecount = 0;
while ((line = br.readLine()) != null) {
characterCount += line.length();
String words[] = line.split("\\s+");
wordCount += words.length;
String sentence[] = line.split("[!?.:]+");
sentenceCount += sentence.length;
System.out.println(++linecount +" "+line);
}
System.out.println("Total word count = "+ wordCount);
System.out.println("Total number of sentences = "+ sentenceCount);
System.out.println("Total number of characters = "+ characterCount);
}
}
```

Operations with Binary files

Q3: Write a Java program to perform the following operations with binary files.

a) Create a folder , section name as folder name in c drive by passing the folder name at run time using Scanner.

b) Create a file to write about students those who submitted above essay , file name must be “sectionname_students.dat”.

c) Add the content to the above file as follows:
The first line is the header line, the remaining lines corresponds to rows in the table, The elements are separated by spaces.

Name	Regdno	Essaysubmitted
Irfan	123	yes
Manoj	124	yes
Pavan	126	no

d) Read the above file to console

Solution :

a) Creating folder

```
import java.io.*;
import java.util.Scanner;
public class CreateFolder
{
public static void main(String arr[])
{
Scanner sc = new Scanner(System.in);
System.out.println("enter folder name");
String foldername=sc.next();
File f=new File(foldername);
if(f.exists()&&f.isDirectory())
System.out.println("already exist");
else if(f.mkdir()==true)
```

```java
System.out.println("successfully created");
else
System.out.println("cannot be created");
}
}
```

b) Creating file

```java
import java.io.*;
import java.util.Scanner;
public class Main {
public static void main(String args[]) throws Exception {
Scanner sc = new Scanner(System.in);
System.out.println("Enter file name and folder name");
String filename=sc.next();
String foldername=sc.next();
FileOutputStream output = new
FileOutputStream(foldername+"\\"+filename+".dat");
System.out.println("file created");
String header = "Name\tRegdno\tEssaysubmitted";
byte[] arrheader=header.getBytes();
output.write(arrheader);
System.out.println("file written");
output.close();
}}
```

c) Add content to file

```java
import java.io.*;
import java.util.Scanner;
public class Main {
public static void main(String args[]) throws Exception {
Scanner sc = new Scanner(System.in);
FileOutputStream output = new FileOutputStream("4cse\\4cse_student.dat",true);
String record=new String();
String name;
int regdno;
char submitted,ch='n';
do {
System.out.println("Student Name , regdno in integer, submitted or not (y/n)");
name=sc.next();
regdno=sc.nextInt();
submitted=sc.next().charAt(0);
String
newrecord=record.concat("\n").concat(name).concat("\t").concat(String.valueOf(regdno)).concat("\t\t").concat(String.valueOf(submitted));
byte[] arrheader=newrecord.getBytes();
output.write(arrheader);
System.out.println("enter y for more students and n to stop");
ch=sc.next().charAt(0);
}while(ch!='n');
System.out.println("file written");
output.close();
}}
```

d) Reading File to console

```
import java.io.*;
public class ReadBinaryFile {
public static void main(String[] args) throws Exception {
File file = new File("linus\\4cse_student.dat");
FileInputStream fis = new FileInputStream(file);
BufferedReader br = new BufferedReader(new InputStreamReader(fis));
String line;
while ((line = br.readLine()) != null) {
System.out.println(line);
}
}
}
```

Operations with Channel and Buffer

Buffers work with the channel. Channels are the tube through which data is transferred and buffers are the source and target of those data transfers. In the case of a write, data we want to write is placed in a buffer, which is passed to a channel then the channel reads that data from the buffer and writes it into the file.

Q4: Write a Java program to perform read and write operation with a data file using FileChannel and ByteBuffer.

Solution :

```
import java.io.*;
import java.nio.*;
import java.nio.channels.FileChannel;
```

```java
public class FileChannelDemo {
public static void main(String args[]) {
String data="iphone 6 50000";
write("tablet.store", data);
read("tablet.store");
}
public static void write(String filename, String data) {
try {
RandomAccessFile store = new RandomAccessFile(filename, "rw");
FileChannel channel = store.getChannel();
ByteBuffer buffer = ByteBuffer.allocate(data.length());
for (int i = 0; i < data.length(); i++) {
buffer.put((byte) (data.charAt(i)));
}
// Rewinds buffer, the position is set to zero
buffer.rewind();
channel.write(buffer);
channel.close();
store.close();
}catch(Exception e) {}
}
public static void read(String filename) {
try {
RandomAccessFile store = new RandomAccessFile(filename, "rw");
FileChannel channel = store.getChannel();
ByteBuffer buffer = ByteBuffer.allocate(1024);
int numOfBytesRead = channel.read(buffer);
System.out.println("number of bytes read : " + numOfBytesRead);
for(int i=0;i<numOfBytesRead;i++) {
char c=(char)buffer.get(i);
System.out.print(c);
}
channel.close();
store.close();
}catch(Exception e) {}
}
}
```

Operations with Serialization

Q5: Write a java program to serialize and deserialize a student object under the file name file named student.ser.

Solution :

```java
import java.io.*;
import java.io.Serializable;
class Student implements Serializable {
String name;
String regdno;
double cgpa;
}
public class SerializeDemo {
public static void main(String [] args) throws Exception {
Student s = new Student();
s.name = "sanjay";
s.regdno = "20213cse0123";
s.cgpa = 7.5;
FileOutputStream fileOut =new FileOutputStream("tmp//student.ser");
ObjectOutputStream out = new ObjectOutputStream(fileOut);
out.writeObject(s);
out.close();
fileOut.close();
System.out.printf("Serialized data is saved in /tmp/employee.ser");
}
}
import java.io.*;
public class DeserializeDemo {
public static void main(String [] args) throws Exception {
Student s = null;
FileInputStream fileIn = new FileInputStream("tmp//student.ser");
ObjectInputStream in = new ObjectInputStream(fileIn);
s = (Student) in.readObject();
in.close();
fileIn.close();
System.out.println("Deserialized Student...");
System.out.println("Name: " + s.name);
System.out.println("Regd no: " + s.regdno);
System.out.println("CGPA: " + s.cgpa);
}
```

```
    }
```

Record Writing

Note: Submit the record on or before the due date.

Both Soft and Hard copies are required to be submitted.

RECORD WRITING INSTRUCTIONS

1. Solve the programming exercise using any IDE (Laptop / Mobile) or using any online compiler.
   A. Students can use online compiler or any preferable platform for the execution. Suggested is to use JDoodle. https://www.jdoodle.com/online-javacompiler/ Do test this site before your CA.
   B. Mobile users, kindly install JStudio - ide for java https://play.google.com/store/apps/details?id=com.qamar.ide.java&hl=en. This instruction is already given for solving your lab programs. If you haven’t done, please do install, and test the app as soon as possible.
2. While solving your programming exercise, write the code in A4 sheet paper/Record. While writing on the paper, please add these info. “Presidency University” “Department of CSE” “Odd semester 2021- 2022” “MODULE - 1" “Course code : CSE 3146”, Course name : Advanced Java Programming, ID:_________________________, NAME:_________________________________, SEC:____ , Date:__________________________
   A. While coding (the soft copy) & writing in the paper/record, all your **CLASS NAME** and the **METHOD NAME** must be appended with your **LAST FOUR DIGIT student ID**. This is mandatory, even while WRITING in the paper/record.

   B. Take a screenshot of **your program & the output** from your mobile/laptop.
   C. Take a photo of the handwritten program.
   D. Put together (**4. A,B,C**) , **combine as one pdf**, with the file name as your student **registration number(ex. 2020BCA0161.pdf)**, and upload the file in Camu.
   E. The document must be uploaded within the specified time in Camu.

Kindly follow the instructions very carefully so that your submission will be valid.