CS116

LAB 7

This Lab is to be started in the lab. It is due April 14 on Blackboard time stamped by 10:00 p.m.

This lab. assignment is worth 4 points.

Objectives:

1. Using IO streams.
2. Writing binary formatted data into a file.
3. Reading binary formatted data from a file.
4. Implementing methods that can throw multiple exceptions.

Task 1 (2.0)- Writing/Reading Data in a Binary Form

1. In this lab we are going to create records and record them in binary format in a file. (simulating a database, where data is stored). Binary data is written in non readable by a human eye form in a file (encoded). An example of a record is a line of data that includes many fields and looks like a table. Each row of data in the table is called a record.

   Here is an example of a record for a table that keeps track of Bank Account information:

Bank_Account_Num	First_ Name (first character)	Last_ Name (first character)	Balance	Fees_Apply (true is fees apply, false is fees don't apply)	Password (a String)
1234	A	M	1000.30	true	Mypassword

   The first row of the table defines the names of the fields. Each field must be defined in terms of the data type that pertains to that field. For example the field "Amount of Balance" could be a double data type and Password is a String. The values of each record pertain to a single customer. Multiple customers will be represented by multiple records (multiple rows) in the table.

2. Therefore, a record has fields and the fields have names. The field names correspond to names of the attributes of an object of a class. For example, consider the example of a record shown above. Let us say that the record fields correspond to the attributes of a service class called BankAccount . Each field (attribute) has a data type associated with it.
   Thus, Bank_Account_Num is an int data type, First_Name is a char data type, Last_ Name is also a char data type, Balance is a double data type, Fees_Apply is a Boolean data type and Password is a String.

4. A client class to the service class will create the BankAccount objects. Each object has values for its attributes and therefore those values for each object constitute a record. The bank wants to keep the information secret of course when it records the records in a file. Therefore the bank would want to write the attributes of each object (a record) using binary coded streams. The bank will be reading the recorded records from the file by using the corresponding streams that read binary coded data.

5. **DataInputStream** and **DataOutputStream** are streams used to read and write raw data of bytes (a byte is 8 bits, where a bit is either a binary 0 or a binary 1).
   Both streams have multiple read and write methods, one for each data type as a matter of fact. Look up the API for each of these streams!!!

   The classes FileInputStream and FileOutputStream inherit form InputStream and OutputStream classes respectively. Both sets of classes have methods and constructors like for example(taken from the API):

   FileOutputStream(String filename, boolean bool)
   where the boolean' s value of true indicates that we want the file to be appended. If the file does not exist it is then created. Creates a stream, to connect to the file whose name is held in the String:filename, for writing.

   FileOutputStream(String filename)
   creates a file by the given filename if it does not exist. If it exists it replaces what is already written in it. It overrides existing data in the file.

   FileInputStream(String filename)
   creates a stream to connect to the file given by String:filename, for reading.

   The FileOutputStream object has to be passed as argument to the DataOutputStream constructor. The same is true for the FileInputObject which has to be passed as argument to the constructor of the DataInputStream class.

Before you start the coding part, open the API and study carefully the methods available in all 3 classes (various read and write method

6. PROGRAMMING INSTRUCTIONS:

a) Based on the discussion above, create the service class BankAccount with attributes as discussed in the example above (see the record in the table).
b) Use package Compiled for all classes.
c) Create a class CreateRecords that has a main method.
d) In the main method do the following:
   Create the following BankAccount objects:
   1234 A M 1000.30 true Mypassword
   3456 G L 300.34 false helenB
   7890 H J 1290.0 true jwer
   6781 F D 260.60 true hgfdw

SAMPLE OUTPUT:
Program produces binary coded file records.dat

```
C:\CS116\SPRING2014\Labs\Lab7\Lab7Solution\Task1>java CreateRecords
please indicate if you want the file records.dat read by typing yes
yes
1234 A M 1000.3 true M y p a s s w o r d
3456 G L 300.34 false h e l e n B
7890 H J 1290.0 true j w e r
6781 F D 260.6 true h g f w
```

If you want to look at the contents of the records.dat file, you could opened the file using EditPlus or Notepad++

TASK 2 (2.0 points)

Repeat the above task but this time write each record in a text file (therefore you are writing Strings now) called data.txt. The data should be formatted space wise in the text file so that it resembles a record (use new line character and tab escape characters to position the data in the file).
Escape characters need to be written using the stream object in the form of a String (surrounded by " "), except that the stream recognizes them as formatting symbols rather than a symbol to be actually written .
After writing the data, ask the user if he/she wants to read the data from file data.txt. if yes display the result on a DOS window.
Do not use the scanner object to read the text file but instead use the proper streams.

Write the 4 records into the **text** file : output.txt (make sure that you use tabs and new line escape operators in your writing)

Next, ask the user if he/she wants the records read off the file. If the answer is yes, open the file records .dat and read the records.
Display on the DOS window the records, one record per line as read from the file records.dat.
No tabs should be used in the display!

e) Make sure that you close the streams when done either writing or reading.

The output should look the same as the data provided in paragraph 6d

```
---------------------------------------------------------------------------------------------
C:\CS116\SPRING2014\Labs\Lab7\Lab7Solution\Task2>java CreateRecords

please indicate if you want the file output.txt read by typing yes
yes
1234    A      M       1000.3  true    Mypassword
3456    G      L       300.34  false   helenB
7890    H      J       1290.0  true    jwer
6781    F      D       260.6   true    hgfw
```

Submission instructions

- In your submission you must include
    a. This document with the answers (copies of the outputs as requested above).
    b. The source code files and the compiled files from Task 1 in the folder Task1.
- Zip all files and name the zip file using your first name followed by your last name followed by lab1.
    i.e. George_KayLab7.zip
- Upload the file on Blackboard by 10:00 p.m. on the due date (Blackboard time stamps automatically)

SUBMISSION INSTRUCTIONS
Submit in assignment folder Lab 7.
Make sure that each task is in its own subfolder in the submission and that the source code files are included.