**Text File I/O Assignment**

**Knowledge (10 marks)**

**1. Why would you want to store data in a file? (1)**

We store data in files because we want to be able to use that data each time we run our program. If a file is not used to store the data, the data would be erased each time the program is closed. By using a file, the program can write information to a file. The program can then access the data by reading the file and searching for the required data.

**2. Compare/Contrast saving information in a text vs binary file. (3)**

Both text and binary files store data as a series of bits. To obtain information, a program reads a ‘stream’ of bits. Text file represent bits as characters from the ASCII table. Text files can only store text data so they are more restrictive. Binary files represent bits as custom data. Binary files use a sequence of bytes or group of order eight bits. A programmer sorts the bytes into a format that will store the information for the program. Binary file formats include many type of data in the same file like video, image, and audio. The data can be read in specific programs, but will appear unreadable in a text editor. The classes used to writer to the different files are different. To write to a binary file the following output streams are used: OutputStream, BufferedOutputStream, FileOutputStream, and DataOutputStream. On the other hand, text files use the following output streams: Writer, BufferedWriter, OutputStreamWriter, FileWriter and PrintWriter.

**3. What class(es) are used to read from a text file? (list all discussed in class) (2)**

Below are the steps to read from a file:

1) Create File object: The File Class gives you access to a file in a specific directory. File acts as a class representing a file location and name.

2) Create a Scanner for the file: We then use the Scanner Class. Simple input is accomplished by using a scanner to scan the files for lines, text, etc (it actually scans for ‘tokens’). The default ones include line, int, word or you can use a ‘delimiter’.

3) Read all your data: Search and read through the file to obtain the data you need.

4) Close the Scanner

**4. What class is used to write to a text file? (list all discussed in class) (2)**

Before writing to a file, a file object is created using the File class and make sure to import java.io\* and throw IOException or use try catch. The PrintWriter and FileWriter Class allow us to write numbers and text to files. After creating it on a specific file, we only need to use the methods to write the data. The FileWriter class lets us write characters, strings, ints to a file using the ‘write’ method. We use the PrintWriter method because it uses FileWriter, but also adds convenient methods we are use to using such as println(…) and print(…). Note that the default behaviour of the class is to overwrite everything. Therefore use a different constructor class to append to a file using: new FileWriter(String fname, boolean append). Also make sure to close the PrintWriter.

**5. With modern encryption (like sha256), is it possible for a program or website to**

**retrieve a forgotten password? (1)**

Modern encryption such as sha256 use a method called one-way encryption. This means that the password can be encrypted but not decrypted. For example whenever a password is created for an account, the password is encrypted using the algorithm and then it is stored in the file. When you want to log in to the account, the password used to log in is also encrypted and then compared to the password on the file. Therefore the program never actually knows the real password. Unless someone were to find a bug in the algorithm, it is not possible for a program or website to retrieve a forgotten password. It could retrieve the encrypted password but not the original password.

**6. Can binary files be viewed properly in a text editor. Explain your answer (2)**

When a binary file is viewed using a text editor, a group of 8 bits will usually be translated to a single character. The person viewing the file would see data that is unreadable. The person would see random characters in the text editor and would not be able to read the data. Numbers in a binary file are not stored as character representations of their digits but as their values encoded in binary. Also a file could become corrupt if a binary file is saved in a text editor because the byte sequences change. Binary files need a specific software to read and write from it. To summarize, although the user would be able to see information, they would not be able to view it properly in a text editor.

**Thinking (6 marks)**

**1. Explain how you might work with a list of books in a library in which you want to**

**store many different pieces of information about each book item in a single file.**

If you want to store a list of books with information, you should store the information using a delimiter. Firstly, treat each line in the file as a single book. When writing to the file, use a delimiter to separate the difference fields such as title, author, published date, etc. This makes it easier to access the required information when reading from the file. When you read a line, you will be able to split the line and place each piece of information about a book in an array. Then you can access all the information of that book using that array and its position.

**2. How would you handle reading delimited text where the delimiter might appear**

**as part of the fields themselves (so ‘,’ separates text but could also be in text that**

**should be kept together. Example: I am one,text,I’m the second,I’m the third**

One solution is to use a delimiter that will likely not appear in your text. This can be done using a series of characters. However this solution is not the best because there is still a chance that it could appear in the text. This solution does not solve the problem. The following are more format ways to deal with delimiter collision. One way to solve this problem is to use ASCII delimited text. From 28 to 31 on the ASCII table, there exists non-printing characters that can be used as delimiters. To separate fields, the unit separator (ASCII 31) can be used. Another way to solve this problem is to use regular expression. This is a way of doing searching, editing, and replacing operations on strings. A regular expression is like a special text string that describes a search pattern. This pattern is used to match character combinations in strings.

**3. In many cases, files are related to one another. For example a library patron info might be stored in one file, and a list of borrowed books in another. How would you connect the information between the files?**

To connect the information between the files, you would need to keep something common between to the two files. This can be done using an ID field in the file. For example, in the file with the list of borrowed books, you give each book its own ID. Since each line is one book with delimited text, each line will have its own ID. The other file with the library patron info will use the same ID for the corresponding patron. Therefore, you will be able to access both file's information using that one ID.