ICS 102 LAB#09 – UPDATING TEXT-FILES

Text-files are not random-access structures. To update a text-file, the following three techniques can be used:

**Updating a small text-file using an array and by overwriting:**

1. Create a **Scanner** object to read from the file.
2. Create an array of appropriate type. You may need to count the number of “items” in the text-file to obtain the size of the array.
3. If you counted the number of “items”, reset the Scanner object to point to the beginning of the text-file.
4. Initialize the array with the file “items”.
5. Close the input stream.
6. Update the array with required modifications.
7. Create a **PrintWriter** object to write to the file.
8. Overwrite the file with the values of the updated array. Make sure not to generate an extra blank line at the end of the updated file.
9. Close the output stream.

Examples:

* A text-file contains an unknown number of integers. Write a Java program that will update the text-file by adding two to each integer in the file each time it is run. [See **FileUpdate1.java** and **input.txt**]
* A text-file contains an unknown number of names. Write a Java program to modify the **kth** name in the file by another name, where **k** is an integer read from the keyboard. [See **FileUpdate2.java** and **names.txt**]

**Updating a small text-file using a string:**

1. Open the file for reading and then read the contents of the file into a string.
2. Update the string with the required modifications.
3. Open the file for writing and overwrite it with the values of the updated file.
4. Close the output stream

Example: A text-file grades.txt contains the integer grades of students. Write a Java program that will update

the text-file by changing each grade of **82.0** to **95.5** [See **UpdateByString.java** and **grades.txt**]

**Updating a small text-file using a temporary file:**

1. Create a **Scanner** object to read from the input file.
2. Create a temporary output file.
3. Create a **PrintWrite**r to write to the temporary output file.
4. Read an item [character, word, number, line, etc.] from the file and modify the item as required.
5. Write the modified items and those that have not been modified to the temporary output file.
6. Close the **PrintWriter** and the Scanner.
7. Delete the input file.
8. Rename the temporary output file as the input file.

Examples:

* A text-file has a single character. Write a Java program that will update the character by incrementing it by 1 [See **UpdateCharacter.java** and **input2.java**]
* Each line of a text-file grades.txt contains the ID of a student followed by his five quiz grades. Write a Java program that will modify a given quiz grade for a given student. [See **UpdateQuizScores.java** and **scores.txt**]