

Homework #9
(Java Programming for Beginners - OnLine)
(Directly helps with your final)

There are three distinct areas you need to research and focus on to get your final done properly:

- Process command line arguments to get the names of input and output files from the user using Elipse
- Read and write information from and into the disk files
- Parse the data read from input file, store in a variable, process it and then print it on the screen and into the output file.

Here are couple of exercises, which will help you in writing code for the above problems.

9.1 Write a class named *Exercise5*. Your main method's header looks like this:

```
public static void main(String[] args)
```

where, *args* is the *String* array of command line arguments passed by JVM. Write a method *processCLArguments()* like this:

```
public static void processCLArguments(String[] args)
```

if the command *args* contains less than two strings, it displays the message like this:

```
Usage: java Exercise5 inputFile outputFile
```

If the arguments contains input and output file names it displays the message like this:

```
Input will be read from: input_final.txt
```

```
Output will be written into: output_final.txt
```

9.2 Write another static method called *processInputOutputFiles()*, which takes a *String* array as arguments which has names of input and output files.

```
public static void processInputOutputFiles(  
                                           String[] args)
```

It reads the content of the input file whose name is in the first element of *args* *String* array with the scores for unknown number of students like the following. The format of the data is, name of a student, followed by comma-separated scores of *quiz1*, *quiz2*, *quiz3*, *quiz4*, *midterm 1*, *midterm 2* and *final* for that student:

```
Thui Bhu, 100, 90, 80, 100, 89, 99, 88
Ariana B. Smith, 90, 90, 100, 100, 99, 100, 95
//more students data
```

Read each line (scores of a student) at a time, and

a) display the output on the screen like this:

```
Student #: 1 Thui Bhu, 100, 90, 80, 100, 89, 99, 88
Student #: 2 Ariana B. Smith, 90, 90, 100, 100, 99, 100, 95
```

b) and print the output on the disk file whose name is given in the second element of args String array. The disk output looks like this:

```
Student #1 is: "Thui Bhu" whose raw scores are: 100: 90: 80: 100: 89: 99: 88:
Student #2 is: "Ariana B. Smith" whose raw scores are: 90: 90: 100: 100: 99: 100: 95:
```

9.3 Research on how to give the command line arguments in your class file using Eclipse.

When starting a Java application, you may want to supply few input parameters to your application even before your program starts. For example, giving database names, supplying username and password etc.

Supplying these arguments before the application starts differs in the way you start your application. You can run your Java application through command window (Windows) or terminal window (MAC). You can even supply these arguments while running from your favorite IDE, e.g. Eclipse.

So, there are two questions:

- a) How do you supply these arguments to your program, and
- b) How do you process these arguments in your program