1) **P3A1\_SIVARAMANVENKATARAMAN\_aswinras**

While running this java file, please the input source file as command line argument and to see the output of this file give a > symbol and give the output file name.

**Compilation: javac** **P3A1\_SIVARAMANVENKATARAMAN\_aswinras.java**

**Run it using: java P3A1\_SIVARAMANVENKATARAMAN\_aswinras emacs.log > output.txt**

The purpose of this class is as follows:

1) Read the input log file and extract content from it

2) The content can be anything such as number of revisions for a file or the file having highest number of users

3) This file extracts all this content based on pattern matching, since the input file has repeated patterns occurring. For e.g. the user name always occurs near “user:” text. Similarly, the revision id occurs near

the text “revision id:” etc.

4) So, after parsing, the contents with respect to the file are stored in a file object. Storing it in file object, makes it easy to write to a file and sort data among the files i.e. the file with highest number of users, file with most commits etc.

5) **This class can be extended even to read machine generated log, which are meant for monitoring environments. The data collected from sensors are usually in the form of logs. So, using this data as source, we can extend this class to find, the sensor that gets repair frequently. The sensor that frequently records abnormal temperatures.**

6) **This log data analysis can also be used to monitor user behavior in a website i.e. the users spend more time in which part of the website. This can also be used to detect fraudulent transactions etc**.

2) **P3A2\_SIVARAMANVENKATARAMAN\_aswinras**

**Instruction to compile: javac P3A2\_SIVARAMANVENKATARAMAN\_aswinras java -cp opencsv-3.10.jar**

**Instruction to run: java -cp .;opencsv-3.10.jar P3A2\_SIVARAMANVENKATARAMAN\_aswinras**

1) Though like the earlier class, we do log data analysis, we write the analyzed data to excel using this class.

2) The CSVWriter object of opencsv class is used to perform this action

3) **We can perform more complex data analysis on the data and feed to Microsoft Excel, for it to perform intuitive visualizations. It can include analysis of number of commits by the user made in a collaborative environment Day-Wise, Week-wise, Month-wise etc. So, generally in organizations, the commits made by users tend to be higher during production phase, so all these data can be analyzed using java and sent to Microsoft excel for visualization**.

4) So, this connection between Microsoft excel and java can create world opportunities in analytics. In this scenario, Java has acted as a querying language to perform analysis on data .

3) **P3A3\_SIVARAMANVENKATARAMAN\_aswinras**

**Instruction to compile: javac P3A3\_SIVARAMANVENKATARAMAN\_aswinras java -cp opencsv-3.10.jar**

**Instruction to run: java -cp .;opencsv-3.10.jar P3A3\_SIVARAMANVENKATARAMAN\_aswinras**

1) This class is used to generate report for time chunks.

2) This report should consist of top 20% users in terms of number of commits

3) The solution I have implemented finds the top 20% users in terms of number of commits, year wise from 2001-2015.

4) The solution can be extended to find the highest number of commits made quarterly, half yearly for every individual year etc.

5) The solution can be extended to find the top 20% users in terms of commits given a date range