**Instruction for compiling and running Homework 4**

* Files: Zip file comes with a JAVA project exported directly from eclipse and contains following files:
  1. *Source code of Lucene Implementation in Java:* following files can be found under src folder
     1. *LuceneImplementation.*java (Main class)
     2. *TermFrequencyBuilder.*java (Used for building term freq dictionary)
     3. *IndexFiles.*java (indexes the files)
     4. *ValueComparator.java (For comparing and sorting dictionary)*
  2. *Corpus*: the corpus (html) files provided for indexing and ranking present under corpus folder inside the project.
  3. *queries.txt*: the query file containing 4 queries required to score documents.
* Steps to run HW\_4 Assignment:

1. Extract the Archived Project and Navigate to HW4 folder then run below command:

> java -cp lib/\*:bin/ LuceneImplementation

* Libraries Used:
  1. *lucene-analyzers-common-4.7.2.jar*
  2. *lucene-core-4.7.2.jar*
  3. *lucene-queryparser-4.7.2.jar*
  4. *jsoup-1.8.3.jar* (Used for filtering out the HTML tags from the corpus before indexing)
  5. *xchart-2.6.0.jar* (Used to plot the graphs from the program)
* Outputs:

1. *term\_frequency*.txt: (terms indexed and their frequency in descending order)
2. *Rank vs Probability.jpg:* (Graph generated to verify zipf’s law)
3. *Log Rank vs Log Probability.jpg:* (log-log graph)
4. *q1, q2, q3 & q4:* (top 100 documents along with scores for each query)
5. *ComparisonBM25vsLucene.xlsx:* (Comparison of BM25 and Lucene Search results)

***References***:

* Stackoverflow: <http://www.stackoverflow.com/>
* Lucene Docs: <https://lucene.apache.org/>
* XChart: <http://knowm.org/open-source/xchart/>

Note:

1. I have provided the output files generated by running the program LuceneImplementation.java in this zip file for your reference but in actual these output files are generated at runtime as per given corpus input.
2. The path of index location, corpus files and query file are taken as constant in the program for this assignment but it can be changed to accept inputs from command line in future with minimal efforts.