**CSEE5590-0001/490-0003: Big Data Programming**

**Lesson Plan # 2**

**ICP Feedback and Submission Link :** <https://docs.google.com/forms/d/e/1FAIpQLSe1WigyMHd3SVLiynDleH1Njm0DWp8skiYBfE-ZJx6HhflG0Q/viewform>

**For Online students:** <https://docs.google.com/forms/d/e/1FAIpQLSe1WigyMHd3SVLiynDleH1Njm0DWp8skiYBfE-ZJx6HhflG0Q/viewform>

**Lesson Title:** Hadoop MapReduce and Hadoop Distributed File System (HDFS)

**Lesson Description:** In this lesson, we are going to discuss about Hadoop MapReduce and Hadoop Distributed File System (HDFS)

**In class exercise**

**There are many ways to execute wordcount program:**

1. **Using any IDE like Intellij or Eclipse**
2. **Run on hadoop clusters**

**Use case Description:**

A screenshot of a cell phone

Description generated with high confidence

1. Counting the frequency of words in the given input with MapReduce algorithm

Use the following text file to count the frequency of words.

<https://umkc.box.com/s/r4jtmjnoip7g0q8tzyqb2naa78u50t3c>

Refer the following link for step by step explanation of wordcount program runs on single node cluster.

<https://github.com/chenmiao/Big_Data_Analytics_Web_Text/wiki/Hadoop-with-Cloudera-VM-(the-Word-Count-Example)>

1. Counting the frequency of words in given text file that starts with letter ‘a’

Use the text file as input.

<https://umkc.box.com/s/r4jtmjnoip7g0q8tzyqb2naa78u50t3c>

Refer following example:

**Input text: ashes**

**time**

**cat**

**add**

**add**

**amazing**

**Output: ashes 1**

**add 2**

**amazing 1**

**Bonus Question:**

**Determine the prime number in input and print number only once**

**Input:**

**2**

**3**

**3**

**7**

**7**

**6**

**8**

**Output:**

**2 0**

**3 0**

**7 0**

**6 1**

**8 1**

**0 à Prime**

**1 -> Not Prime**

Marks will be distributed between logic, implementation and UI

**Programming elements:**

Hadoop MapReduce and HDFS

**Source Code:**

<https://umkc.box.com/s/x89szyb48chr4r1zm0fp0plafczi6ogx>

**Prerequisites:**

Ensure that Hadoop is installed, configured and is running. More details:

[Single Node Setup](https://hadoop.apache.org/docs/r1.2.1/single_node_setup.html) for first-time users.

[Cluster Setup](https://hadoop.apache.org/docs/r1.2.1/cluster_setup.html) for large, distributed clusters**.**

**You can even execute program on any IDE like eclipse or Intellij.**

After completion of your ICP fill in the form. Any available TA/instructor will come to you and evaluate ICP

**ICP Guidelines (In Class Students):**

1. ICP Submission is in pairs of two students.
2. Once completed, must be presented to TA or Instructor before the completion of the class
3. Submission after class is considered as late submission. (Check the late submission policy in the syllabus)
4. ICP Code with brief explanation should be pushed to GitHub. Submit Github Link through the Feedback Form (<https://docs.google.com/forms/d/e/1FAIpQLSe1WigyMHd3SVLiynDleH1Njm0DWp8skiYBfE-ZJx6HhflG0Q/viewform>**)**

**Submission Guidelines (for online students):**

1. Submit your source code and documentation to GitHub and represent the work through wiki page properly (submit your screenshots as well. The screenshot should have both the code and the output)
2. Comment your code appropriately.
3. Submit a brief demo video 2-3 min showing your assignment with a voice over explaining your work through the Submission Link.
4. Use the following google link to submit your assignment

(ICP Submission Link#): <https://docs.google.com/forms/d/e/1FAIpQLSe1WigyMHd3SVLiynDleH1Njm0DWp8skiYBfE-ZJx6HhflG0Q/viewform>

***Cheating, plagiarism, disruptive behavior and other forms of unacceptable conduct are subject to strong sanctions in accordance with university policy. See detailed description of university policy at the following URL:*** [*https://catalog.umkc.edu/special-notices/academic-honesty/*](https://catalog.umkc.edu/special-notices/academic-honesty/)