**Aaron Stefano F. Bonaobra**

**BSIT – 3A, Big Data Analysis**

**LAB 1 ACTIVITY DOCUMENTATION:**   
  
**Activity:** Creating and Executing a Spark RDD Pipeline with Five Transformations

**Tools:** Programming Language: Python, Framework: Apache Spark (PySpark), Development Environment: VSCode (with Jupyter Notebook or Python Script)

Steps to Execute the Spark RDD Pipeline:

* **Install PySpark:** “pip install pyspark”
* **Import Libraries:** “from pyspark import SparkConf, SparkContext”
* **Initialize Context:**

“conf = SparkConf().setAppName("Simple RDD Example").setMaster("local")

sc = SparkContext(conf=conf)”

* **Create an RDD from a Python List:** “data = ["Apple", "Banana", "Cherry", "Apple", "banana", "Cherry", "APPLE", "banana"]

rdd = sc.parallelize(data)”

* **APPLY 5 TRANSFORMATIONS:**

1. **Convert all to lowercase:** “lower\_rdd = rdd.map(lambda word: word.lower())”
2. **Filter words with more than 5 letters:** “filtered\_rdd = lower\_rdd.filter(lambda word: len(word) > 5)”
3. **Map words to key-value pairs (word, 1):** pairs = “filtered\_rdd.map(lambda word: (word, 1))”
4. **Reduce by key to count word occurrence:** word\_counts = “pairs.reduceByKey(lambda a, b: a + b)”
5. **Sort words by frequency in descending order:** sorted\_counts = “word\_counts.sortBy(lambda pair: pair[1], ascending=False)”

* **Perform actions and display results:**

“results = sorted\_counts.collect()

for word, count in results:

print(f"{word}: {count}")”

* **Stop spark context:**

“sc.stop()”

**Conclusion:**

This activity demonstrates how to create and execute a Spark RDD pipeline using five transformations. It covers fundamental Spark operations such as map(), filter(), reduceByKey(), and sortBy(), providing a solid foundation for working with big data processing in PySpark.