

## Big Data Lab Answer Sheet.

Please complete this answer sheet and turn it in at the beginning of class on the due date posted in LEARN.

### Part I

<b>Part 1:</b>	<b>Answer</b>
<b>1</b> (4 pts)	Json lays out the information like a dictionary labels its information. It's separated to two columns. The left side is the key and the right side is the value.
<b>2</b> (4 pts)	XML is formatted similar to a html file. Every content is in a tag, with subtasks specify the contents a bit more clearly
<b>3</b> (4 pts)	2015 Alabama: 2552 2015 Alaska: 288 More people live in Alabama so there is more chance for people to die by accident
<b>4</b> (6 pts)	I looked at the US food import, the information and data was fascinating. I prefer JSON, I think it's formatted more nicely.
<b>Part 2:</b>	<b>Answer</b>
<b>5</b> (2 pts)	It prints the first line "hope is the thing with feathers" and counts how many times each word appears
<b>6</b> (2 pts)	Each piece is one line of the poem.
<b>7</b> (2 pts)	Counting the variables for each line is faster than counting the variables for the whole poem
<b>8</b> (2 pts)	Instead of counting how many times each word appears, it changes to count how many times each character appears.
<b>9</b> (2 pts)	Each line is splitted into words. We create an empty dictionary, then we use a for loop to loop through all the words. If the word is already in the dictionary, then add the count by one, if it's not, store in the dictionary and count the value

<b>10</b> (2 pts)	The reducer uses each word and its value from the mapper as an input and emits the number of times the word is used in the poem
<b>11</b> (5 pts)	The first example would count the amount of times a word appears in the line and then return emitted it. This example instead created more instances of the word in exchange for not counting them
<b>12</b> (15 pts)	Alice: 81.67 Bob: 68.0 Carol: 67.0 Dave: 78.0 Eve: 63.67
<b>13</b> (20 pts)	<p>Example 2 Student Scores:</p> <p>Mapper:</p> <pre>def mapper(key, value):     grade_map = eval(key)     for student in grade_map:         grade = grade_map[student]         Wmr.emit(student,grade)</pre> <p>Reducer:</p> <pre>def reduce(key, values):     sum = 0     count = 0     for value in values:         sum = sum + float(value)         count += 1     if count &gt; 0:         average = sum / count         Wmr.emit(key,average)</pre> <p>OUTPUT:</p>

Alice 81.6666666667  
Bob 68.0  
Carol 67.0  
Dave 78.0  
Eve 63.6666666667

EXAPLE 3 Enrollment:

```
{ 'Name': 'ARISE Academy Charter High School',  
'Type': 'CS', 'Enrollments': '183', 'Male Dropouts': '1',  
'Female Dropouts': '1', 'Dropouts': '2' }  
{ 'Name': 'ASPIRA Bilingual Cyber Charter School',  
'Type': 'CS', 'Enrollments': '57', 'Male Dropouts': '2', 'Female  
Dropouts': '6', 'Dropouts': '8' }  
{ 'Name': 'Ad Prima CS', 'Type': 'CS', 'Enrollments': '26',  
'Male Dropouts': '0', 'Female Dropouts': '0', 'Dropouts': '0' } {  
'Name': 'Alliance for Progress CS', 'Type': 'CS',  
'Enrollments': '24', 'Male Dropouts': '0', 'Female  
Dropouts': '0', 'Dropouts': '0' }  
{ 'Name': 'Philadelphia City SD', 'Type': 'SD',  
'Enrollments': '63983', 'Male Dropouts': '3092', 'Female  
Dropouts': '2644', 'Dropouts': '5736' }
```

Mapper:

```
def mapper(key,value):  
    grade map = eval(key)  
    for student in grade map:  
        if student == "Enrollments" or student == "Male  
Dropouts" or student == "Female Dropouts":  
            grade = grade map[student]  
            Wmr.emit(student,grade)
```

Reducer:

```
def reducer(key,values):  
    count = 0  
    for value in values:  
        count += int(value)  
    Wmr.emit(key,count)
```

Outpout:

Enrollments	65273
Female Dropouts	2651
Male Dropouts	3095