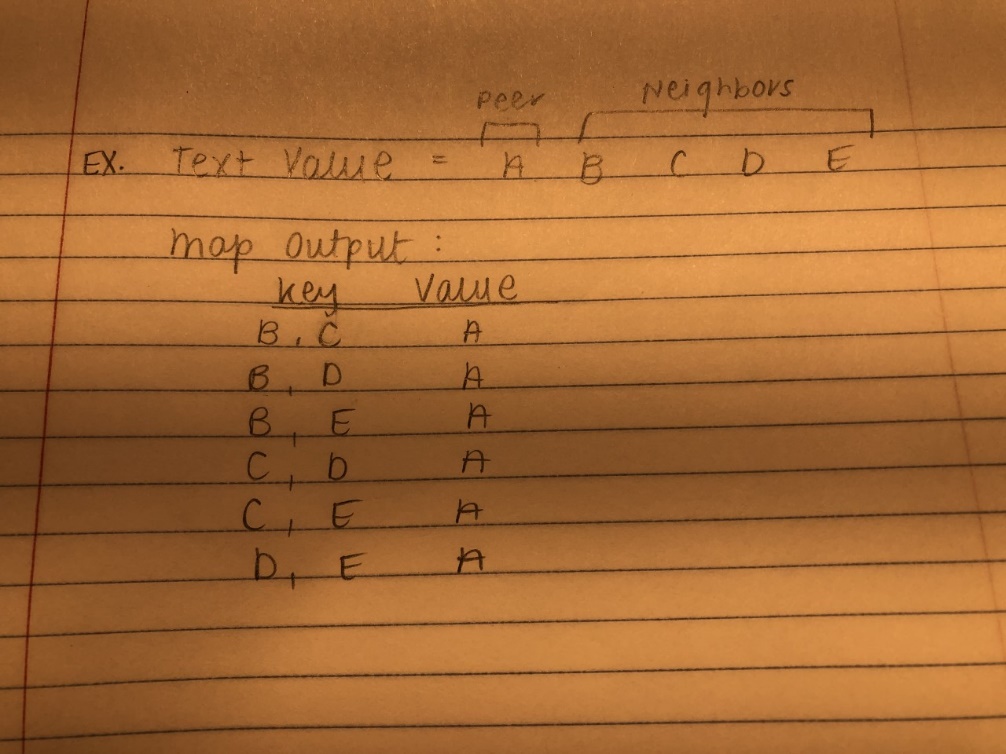
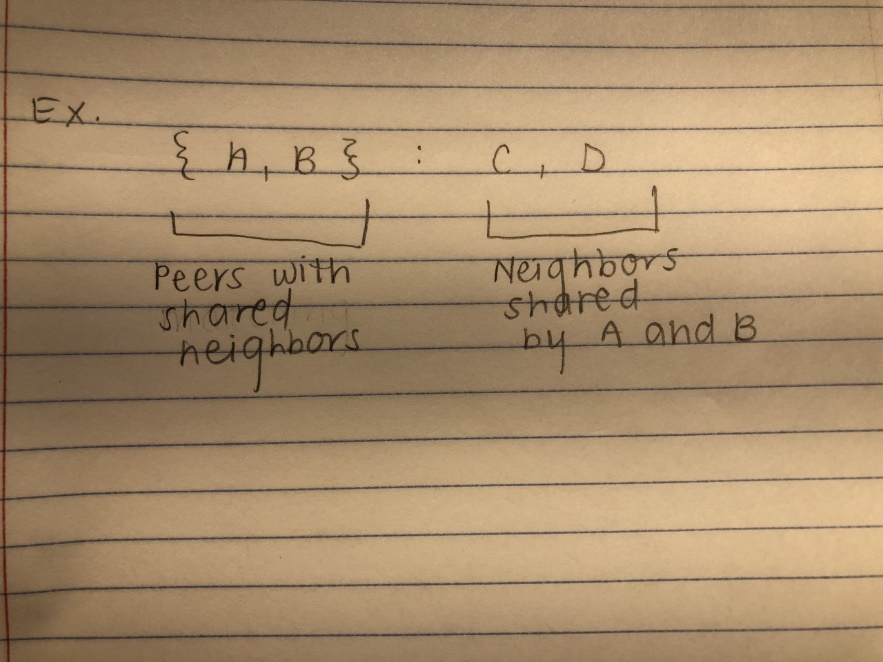
**Implementation**

I implemented the project as follows:

1. Mutual Neighbors class: This is the main class that contains the following classes and methods:
   1. Main Method: This method acts as the driver and will start the program. I also added the appropriate Hadoop commands to set classes, to set output values, and to set the input and output paths.
   2. Map Class: This class will map the data from the input file.
      1. Map Method: The data from the input file is passed to this method through the parameter Text value. From there, I split the data by spaces and put each separate value into an array called lineArray. The first value of the array is the peer. I then looped through the rest of the values in the array to create a new array containing only the neighbors. After that, I created a nested for loop which creates pairs of neighbors. Finally, I wrote this to context, ensuring the first parameter, valueText, is one of the pairs of neighbors, and the second parameter, peerText, is the peer of the line of data. The following image shows my implementation. The “key” in this image is the first parameter, valueText, that I wrote to context. The “value” in the image is the second parameter, peerText, that I wrote to context.



* 1. Reduce Class: This class will reduce the data from the map class.
     1. Reduce method: This method receives data from the map. This data includes a Text key and an Iterable list of type Text called values that contains all the values from the map that had that same key. From there, I simply looped through all the values and formatted them with curly braces, commas, and a colon. Finally, I wrote this to context, ensuring the first parameter, result, is the list of peers who share neighbors, and the second parameter, key, is the list of neighbors shared by the peers. Since there was no instruction on how to structure the output, I chose to structure mine as follows: