**Part A.**

The following is to be word processed

Design a table which will hold the data set. When all data is entered, the table must be greater then 80% full and less then 90%. The table will need to implement insertion, deletion and find.

**What is the size of the table? Prove it?**

The Hash table size should be around 121 to be greater than 80% and less than 90%. The reason why it should be around 121 is because there are 188 words, that’s including duplicate words as well. When not counting the duplicate words, there are 101 words. I used my assignment 3 to check for how many words in the AVL tree. Since the AVL tree doesn’t insert duplicate values, I can count the vertices of the tree. I multiply the numbers of vertices by \*1.20 to make the table 20% bigger to get the table to be around 80% load factor.

**What fields will the table need to represent?**

The hash table will need an array field and tableSize field. I created a Word class which has a String field as key and duplicateCounter field which is an int. The Word class is going to be store in the array field of the Hash table, which is going to be an Word array.

**Give the logical structure of the table.**

**Design a primary hash function which will distribute the keys (strings) as evenly as possible. Can you prove that the distribution adheres to the principles of good hash functions?**

To get a really good string hash function, multiplying the char of the string by 127 casting it to the ASCII value, and Mod it by a big prime number such as 16908799 and Mod it again by the size of the table.

For double hashing I Mod the hashkey by 7 and subtract 7. This will work with other prime numbers, usually 5 or 7 is a good choice to use.

#### Part B.

Code/ implementation on the next page.