

# HASHTABLES – TABLES O' WORDS (PROJECT 2)

**Lab Description :** Write a hash table. A hash table can be implemented using an array of linked lists. Use an array of Java LinkedLists.

## **Part 1:**

Read in Strings and use them to construct Word objects. Word objects will be organized in the hash table in the appropriate bucket using the hashCode value of each Word.

**HASH FORMULA :** ( # of vowels \* # of letters ) % 10

If the word is BOOT then, BOOT has a hash of  $(2 * 4) \% 10 == 8$ .

This hash table does not store duplicate values.

## **Sample Data :**

```
20
one
two
dog
cat
chicken
pig
owl
jump
run
hop
shortcut
ferret
goat
hootowl
owl
go
alligator
onimonapia
food
a
```

## **Sample Output :**

```
HASHTABLE
bucket 0 onimonapia
bucket 1 hootowl a
bucket 2 ferret go
bucket 3 two dog cat pig owl run hop
bucket 4 chicken jump
bucket 5
bucket 6 one shortcut alligator
bucket 7
bucket 8 goat food
bucket 9
```

## **Files Needed ::**

```
Word.java
WordTester.java
HashTable.java - previously
written
HashTableRunner.java
words.txt
```

## **Part 2 :**

Write your own version of hashCode for the Word class. You can modify the hashCode class for this if needed. Be sure that your resulting hashCode has reasonable distribution across the given indexes.

**\*This approach does not have to be better than the hashCode given in Part 1 but you should be able identify whether it is better or worse and why**