Design Document

Below are the design considerations made during design of hash table for the assignment.

**Hash key/Value**

**Hash key**: String

Hash Key format

<YYYYAAADDDD> where

YYYY - represents the year in which this student joined the university

AAA - a three letter (alphabet) representing degree program

DDDD - a four digit number representing the students roll number

**Hash Value**: float

**Hash Table Size**

Hash table size is taken as 157.

Total no of entries in input.txt is 112. Considering the load factor as 0.75 nest prime no is selected.

**Hash Code Mapping**

Polynomial accumulation technique is used for hash code mapping. It’s been tested that no of collisions using hash code mapping are very negligible when used with a weight 31.

The maximum possible value with weight 31 for the hash key is 1640321990546224 which can be accommodated by java data type long.

**Compression code Mapping**

For compression code mapping Multiply, Add and divide rule is used.

*Hash index = ((a \* hashCode) + b) % N;*

a is chosen as 3 and b as 269 they are random prime no.

N is size of hash table.

**Collision Handling**

Double Hashing technique is used for collision handling. Since double hashing is most effective technique and doesn’t leads to problem like primary clustering and secondary clustering.

How it works:

Once the collision is detected for *i=h(k)* next possible index is calculated as

*A[(i + f(j)) mod N]*

*Where j = 1, 2, 3, ..., where f(j) = j\*h'(k)*

Second Hash function

*h'(k)* *= q - k mod q*

q is selected 101. q can be any prime no such that *q < N*.

Using q as 101 leads to a maximum computation 6 times and average computation <2 times for finding next possible index for storing the records once the collision occurs.