Computer Science 3A

Practical Assignment 9

4 May 2023

Time: 4 May 2023 13:45 – 17:00 Marks: 50

Practical assignments must be uploaded to `eve.uj.ac.za` **before** 17h00 in the practical session.

Late submissions **will not be accepted**, and will therefore not be marked. You are **not allowed to collaborate** with any other student. You **must** upload your assignment to Eve **before** it will be marked.

Hash Tables are a quick and convenient method of implementing a Map ADT. The basic principle of a Map is that items can be added and removed from the map, the items are associated with Key and Value pairs, and that there is no duplication of keys.

You are required to implement a HashTable that makes use of the hash function defined below. This has been defined for a byte array but your implementation should allow strings and integers to be hashed using this function (the Integer and the String should be converted to a byte array). Objects that are not Integers or Strings should make use of the built in Java hashCode function for the object.

```
Input: a byte array
Output: a hash code

hash <- 5381
for each byte b:
hash <- ((hash << 5) + hash) + b
return hash
```

Your implementation will make use of a normal Object array, where each element in the hash table is a an entry. You must complete the classes and methods marked by:

```
//TODO:
```

A test class has been provided to test your implementation. You should not add any extra functions to the provided classes, only complete the methods that have been indicated.

The following files must be submitted to EVE:

1. *studentnumber*_p9.zip

Bonus

For an additional 15 marks make a Double Hashing Based Hashtable which is the same as the *HashTable* class, but uses a Double Hashing based collision handling approach. So it requires a new *remove*, *get* and *put* methods that do NOT make use of a PositionList at each index, but store a single Entry instead.

Marksheet

1. HashTable: remove	[8]
2. HashTable: get	[8]
3. HashTable: put	[8]
4. HashTable: keys	[8]
5. HashTable: values	[8]
6. Compilation and Correct execution.	[10]