

National University of Computer and Emerging Sciences



Laboratory Manual

for

Data Structures Lab

Course Instructor	Ms. Zareen Alamgir
Lab Instructor(s)	Ms. Mamoon Akbar Ms. Ammara
Section	CS-C
Semester	Fall 2022

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objective:

In this lab, students will practice:

1. Hash Maps

Question 1: (HashMap with Separate Chaining)

Implement a Node struct which represents an item in a hash table.

```
template <class v>
struct HashItem
{
    int key;
    v value;
    HashItem * next
};
```

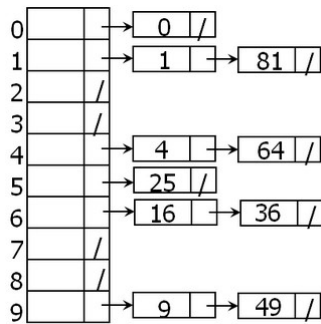
The default value assigned to a HashItem is 0 (empty).

Now implement a HashMap class whose basic definition is as follows (You can add any helping member variables and methods):

```
template <class v>
class HashMap
{
private:
    Vector<v> hashVector;
    int currentElements;

public:
    HashMap();
    HashMap(int const capacity);
    void insert(int const key, v const value);
    bool deleteKey(k const key) const;
    v* get(k const key) const;
    ~HashMap();
};
```

1. HashMap(): constructor creates a hashVector of size 10.
2. HashMap(int const capacity): an overloaded constructor that creates hashVector of size capacity. If capacity is less than 1 return error via assert(capacity>1)
3. void insert(int const key, v const value):
 - a. The insert method inserts the value at its appropriate location. Find the first candidate index of the key using hash function:
 $index = key \bmod capacity$
Please note that the first node on every hash location will be a dummy node.
 - b. In case of hash collision, perform chaining as described in following image



4. `bool deleteKey(k const key) const`: this method deletes the given key. It returns true if the key was found. If the key was not found it returns false. When the key is found, simply set the status of the hashitem containing the key to deleted (value of 1). It also uses status variable to search for the key intelligently.
5. `V* get(k const key) const`: this method returns a pointer to the corresponding value of the key. If the key is not found, it returns nullptr. It also uses status variable to search for the key intelligently.
6. `~HashMap()`: destructor
7. Create a main to test these functions properly.