

lab 18 Hash Tables with Chaining

Instructions: This lab is a practice in constructing a Hash Table with chaining. Implement a Hash Table whose constructor take an integer (the initial size of the hash table), insert, remove, and get. Hints: if the value is not found in the Hash Table return a value using the default constructor. Also, use your previous code!

```
1 #ifndef HASH_TABLE_H
2 #define HASH_TABLE_H
3
4 /* HashTable via chaining */
5 template<class K, class V>
6 class HashTable {
7     private:
8         /* Class to begin filling out...*/
9     public:
10         /* Initialize the Hash Table with size size. */
11         HashTable(const int size);
12
13         /* Destructor shall free up memory */
14         ~HashTable();
15
16         /* Map key -> val.
17          * Return true if sucessful (it is unique.)
18          * Otheriwise return false.
19          */
20         bool insert(const K &key, const V &val);
21
22         /* Print out the HashTable */
23         void print() const;
24
25         /* Remove the val associated with key.
26          * Return true if found and removed.
27          * Otherwise return false.
28          */
29         bool remove(const K &key);
30
31         /* Retrieves the V val that key maps to. */
32         V& operator[] (const K &key);
33 };
34
35 int hashCode(int key);
36 int hashCode(std::string &key);
37
38 #include "hashtable.cpp"
39
```

Write some test cases:

Create some test cases, using `cxxtestgen`, that you believe would cover all aspects of your code.

Memory Management:

Now that are using `new`, we must ensure that there is a corresponding `delete` to free the memory. Ensure there are no memory leaks in your code! Please run Valgrind on your tests to ensure no memory leaks!

STL:

You may use `vector`, `queue/deque`, and `list` from the STL. Do not use any other data structure (especially `map`!) Failure to follow these instructions is an automatic 0 for this lab.

How to turn in:

Turn in via GitHub. Ensure the file(s) are in your directory and then:

- `$ git add <files>`
- `$ git commit`
- `$ git push`

Due Date: November 8, 2017 2359

Teamwork: No teamwork, your work must be your own.