

CMSC 206: Data Structures

Lab #12: BMCHashMap

Write BMCHashMap, an implementation of an associative map based on a hashtable. You may wish to consult BMCHashSet, available on the syllabus page. Your class should implement the BMCMap interface, also available from the syllabus page. Ideally, your class would be declared with

```
public class BMCHashMap<K,V> implements BMCMap<K,V>
```

but if you don't want to deal with generic type variables, you could use

```
public class BMCHashMap implements BMCMap<String, Integer>
```

This last declaration would be a map from Strings to Integers only.

Once you've written your class, write tests to make sure it works.

Extra challenge: Add this method (taken from java.util's Map interface):

```
/** Returns a Set view of the keys contained in this map.
 * The set is backed by the map, so changes to the map
 * are reflected in the set, and vice-versa.
 * The set supports element removal, which
 * removes the corresponding mapping from the map, via
 * the `remove` and `clear` operations. It does not
 * support the `add` or `addAll` operations.
 * @return a set view of the keys contained in this map
 */
public void BMCSet<K> keySet()
```

If your map works only with Strings, then your keySet should return Set<String>. You can experiment with java.util's Map interface and its keyset method to gain a better understanding of what this method should do. It will require writing a new class to implement BMCSet (much like how implementing an iterator method requires writing a new class that implements Iterator).