IT230 Lab 4
Score: _____/10 v1.1

The purpose of this assignment is to learn more about C++ by implementing your own Hashtable.

Specifications

Translate the MyHashtable class made in class to C++.

Your implementation should use a `buckets` array, where each element is an `std::vector<std::tuple<TKey, TValue>>`. The types `TKey` and `TValue` will be class templates. Remember, vectors and tuples are part of the C++ Standard Library (the `std` namespace). Additionally, since arrays in C++ need dynamic sizing, they should be represented as pointers.

Syntax help for tuples: https://www.geeksforgeeks.org/tuples-in-c/#

You must do the following:

- Override the indexing (`[]`) operator.
- Implement both a default constructor and a constructor that takes capacity as an argument.
- Implement an `add()` method to insert items into the hash table.
- Implement a `remove()` method to delete items from the hash table.
- Implement a `clear()` method to reset the hash table.
- Implement a `toString()` method to return a string representation of the hash table.
- Implement a `containsKey()` method that returns true if the specified key exists in the hash table, and false otherwise.

These methods should function the same way as in the original C# implementation. Make sure to test your class thoroughly in the `main()` function.

Constraints

- Utilize a constant for the default capacity. You can define and initialize this in your header file. Have it be type `size_t`.
- The variable that keeps count of how many items are in the Hashtable should be private, however, there should be a public getter for it.
  - You do not need a variable or method for returning how large a bucket is.
- You must not have memory leaks. Utilize a destructor for this.
- Test all of your methods in your main. Make a menu of food items and put them into your hash table the same way we did in C#.
- All methods defined in your source file must have its header in the header file (.h).
- C++ uses camelCase. Name variables and methods using camelCase. Classes should be named with PascelCase. See the tips for help.
- If a method will not affect the state of the object, mark the method as constant.

Submission

The submission for this lab is the commit ID you want to be graded. You will submit the commit ID via the Canvas assignment. If you need help finding the commit ID, ask for help.

10 points will be assigned by correctness and your observable understanding as reflected in your code.

Remember to write efficient code. Optimize your code wherever possible.

Tips

- When trying to get the key from a tuple, use syntax like this:

  ```
  get<0>(buckets[index][i])
  ```

- When trying to get the value from a tuple, use syntax like this:

  ```
  get<1>(buckets[index][i])
  ```

- To check if a vector is null, what you really want to do is check if it is empty by using syntax like this:

  ```
  if(buckets[index].empty())
  ```

- To delete an element from a vector, given index i, use syntax like this:

  ```
  buckets[index].erase(buckets[index].begin() + i);
  ```

- In C++, vectors do not have to be initialized like we initialized the Lists in C#. You can start adding to them right away.
- When you print booleans in C++, you will see ‘1’ for true or ‘0’ for false.
- Remember to add an explicit instantiation of your template class in your class’ source file.