

## Module 7 Assignment: HashMap

🔖 Marcar esta página

### 🔒 Las tareas calificadas están bloqueadas

Cámbiate a la opción verificada para obtener acceso a funciones bloqueadas como esta y aprovechar al máximo tu curso.



Cuando te cambias a la opción verificada, tú:

- ✓ Obtén un **certificado verificado** de finalización para compartirlo en tu curriculum
- ✓ Desbloquea el acceso a todas las actividades del curso, incluidas las **tareas calificadas**
- ✓ **Acceso completo** al contenido y los materiales del curso, incluso después de que finalice el curso
- ✓ Apoye a nuestro **misión** en edX

Opción verificada \$149

Coding Assignment fecha límite Sep 16, 2023 01:05 -05

### HashMap

For this assignment, you will be coding an **ExternalChainingHashMap**, a key-value HashMap with an external chaining collision resolution strategy. A **HashMap** maps unique keys to values and allows **O(1)** average case lookup of a value when the key is known. The table should not contain duplicate keys, but can contain duplicate values. In the event of trying to add a duplicate key, replace the value in the existing (key, value) pair with the new value and return the old value.

#### IMPORTANT:

- You will be given 5 attempts on this assignment, with a 30 minute cooldown between submissions.
- Please run your code before each submission to ensure that there are no formatting errors! If there are formatting errors in your code, your code will not be graded and a submission attempt will be logged. For more information, please review the Vocareum overview below.

### Capacity

The starting capacity of the ExternalChainingHashMap using the default constructor should be the constant **INITIAL\_CAPACITY** defined in *ExternalChainingHashMap.java*. Reference the constant as-is. Do **not** simply copy the value of the constant. Do **not** change the constant. Do **not** regrow the backing array when removing elements.

If adding to the table would cause the load factor (LF) to **exceed** (i.e. greater than, not greater than or equal to) the max load factor constant provided in the java file, the table should be resized to have a capacity of  $2n + 1$ , where  $n$  is the current capacity before adding the parameterized element. See the javadocs for specific instructions on when to resize. There is a method called *resizeBackingTable* that you should use for resizing.

### Hash and Compression Functions

You should not write your own hash functions for this assignment. Instead, use the *hashCode()* method that every Object has. For the compression function, mod by table length first, then take the absolute value (it must be done in this order to prevent overflow in certain cases). As a reminder, you should be using the *hashCode()* method on **only the keys** (and not the *ExternalChainingMapEntry* object itself) since that is what is used to look up the values. After converting a key to an integer with a hash function, the integer must be compressed to fit in the array backing the HashMap.

### Adding

When adding a key/value pair to a HashMap, add the pair to the front of the chain in the correct position. Also remember that keys are unique in a HashMap, so you must ensure that duplicate keys are not added. Each index of the table should point to an *ExternalChainingMapEntry* that represents the head of a linked list. That linked list contains all elements that collide at that index.

## Removing

When removing a key/value pair from a HashMap using external chaining, you can safely remove the item. Removing from a chain is very similar to removing from a Singly-Linked List, treating the first table entry as the head, so refer to your notes and homework assignments from earlier in the course if you need a refresher. As usual, if the entry you are removing is the only entry at that index, you should make sure to null out that spot rather than leaving it there.

### HashMap (External Chaining) Examples

For each example, we assume the hashed index to be  $\text{abs}(\text{key} \% \text{table.length})$ .

Initial HashMap



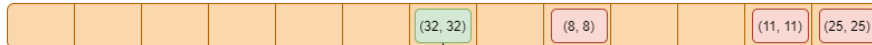
put(5, 5) // If the hashed index is empty, we simply place the new (key, value) pair at that index.



Initial HashMap



put(32, 32) // If there are already element(s) at the hashed index, the new (key, value) pair becomes the head of the chain at that index.



Initial HashMap



put(24, 24) // If the key is duplicate, we simply replace the value and return the old value.



Initial HashMap



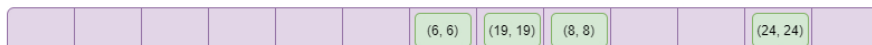
remove(4)



Initial HashMap



remove(11)



### General Tips

- When removing an element from a chain, think about how removing the head differs from removing an element inside the chain, in terms of the backing table.
- We highly recommend copying the starter code and working in your preferred IDE in order to have access to features such as code completion, auto-formatting, and much more!

---

Here are general assignment guidelines that should be followed.

- Do not include any package declarations in your classes.
- Do not change any existing class headers, constructors, instance/global variables, or method signatures. For example, do not add throws to the method headers since they are not necessary. Instead, exceptions should be thrown as follows: `throw new InsertExceptionHere("Error: some exception was thrown");`
- All helper methods you choose to write should be made private. Recall the idea of Encapsulation in Object-Oriented Programming!
- Do not use anything that would trivialize the assignment. (e.g. Don't import/use `java.util.ArrayList` for an `ArrayList` assignment.)
- Always be very conscious of efficiency. Even if your method is to be  $O(n)$ , traversing the structure multiple times is considered inefficient unless that is absolutely required (and that case is extremely rare).
- If applicable, use the generic type of the class; do not use the raw type of the class. For example, use `new LinkedList<Integer>()` instead of `new LinkedList()`.

Use of the following statements should be avoided at all times.

package	System.arraycopy()	clone()
assert()	Arrays class	Array class
Thread class	Collections class	Collection.toArray()
Reflection APIs	Inner or nested classes	Lambda Expressions

---

The Vocareum (code editor) interface has six main components:

- The **Drop-Down** in the top left. This lets you choose from multiple available files. Note that this drop-down will only be visible in assignments that require multiple files.
- The **Run** button. This will compile your code and run a file scan. Running your code will not count towards your total allowed submission attempts, therefore you are free to run as many times as needed.
- The **Submit** button. This will compile your code, run a file scan, grade your assignment, and output results to console. Note that for most assignments in this class, you will only be allowed a limited number of submissions. A submission is counted when the submit button is clicked, regardless of whether or not your code can compile or if there are any file issues. Therefore, we **highly recommend** that you run your code before submitting to ensure that there are no issues that will prevent your code from being graded and that every submission attempt will generate meaningful results.
- The **Reset** button. This will revert all your changes and reset your code to the default code template.
- The **Code Window**. This is where you will write your code. For large coding assignments, we highly recommend copying the starter code and working in your preferred IDE to have access to features such as code completion, auto-formatting, and much more!
- The **Output Window**. This window will appear whenever you run or submit your code and will display the output for you to view.

For additional help, please visit the Vocareum information page located in the course information module!

< Anterior

Siguiente >



## edX

[Acerca de](#)  
[Afiliados](#)  
[edX para negocios](#)  
[Open edX](#)  
[Carreras](#)  
[Noticias](#)

## Legal

[Condiciones de Servicio y Código de Honor](#)  
[Política de privacidad](#)  
[Políticas de Accesibilidad](#)  
[Política de marcas](#)  
[Mapa del Sitio](#)  
[Política de cookies](#)  
[Opciones de privacidad](#)

## Contáctanos

[Centro de ideas](#)  
[Contáctenos](#)  
[Centro de Ayuda](#)  
[Seguridad](#)  
[Kit Multimedia](#)



© 2023 edX LLC. All rights reserved.  
深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)