Day 8: Dictionaries and Maps | HackerRank

Terms you'll find helpful in completing today's challenge are outlined below, along with sample Java code (where appropriate).

Java Maps

Map is an interface that provides a blueprint for data structures that take ($key, value$) pairs and map keys to their associated values (it's important to note that both the *key* and the *value* must be Objects and *not* primitives). The *implementation* is done by *implementing classes* such as HashMap or LinkedHashMap. Consider the following code:

// Declare a String to String map
Map<String, String> myMap;

// Initialize it as a new String to String HashMap
myMap = new HashMap<String, String>();

// Change myMap to be a new (completely different) String to String LinkedHashMap instead
myMap = new LinkedHashMap<String, String>();

Here are a few Map methods you will find helpful for this challenge:

- *containsKey(Object key)*: Returns true if the map contains a mapping for *key*; returns false if there is no such mapping.
- *get(Object key)*: Returns the value to which the *key* is mapped; returns *null* if there is no such mapping.
- *put(K key, V value)*: Adds the (*Key, Value*) mapping to the Map; if the *key* is already in the map, the *value* is overwritten.

Example (Java)

The code below:

// Create a Map of String Keys to String Values, implemented by the HashMap class
Map<String,String> myMap = new HashMap<String,String>();

// Adds ("Hi","Bye") mapping to myMap
myMap.put("Hi", "Bye");

// Print the Value mapped to from "Hi"
System.out.println(myMap.get("Hi"));

// Replaces "Bye" mapping from "Hi" with "Bye!"
myMap.put("Hi", "Bye!");

// Print the Value mapped to from "Hi"
System.out.println(myMap.get("Hi"));

produces the following output:

```
Bye
Bye!
```

It is not necessary to declare *myMap* as type *Map*; you can certainly declare it as a *HashMap* (the instantiated type).

Additional Language Resources

- [Python Dictionary Documentation](#)
- [C++ Unordered Map Documentation](#) and [C++ Map Constructor Documentation](#)
- [C# Dictionary Documentation](#)

[View Practice Challenge](#)