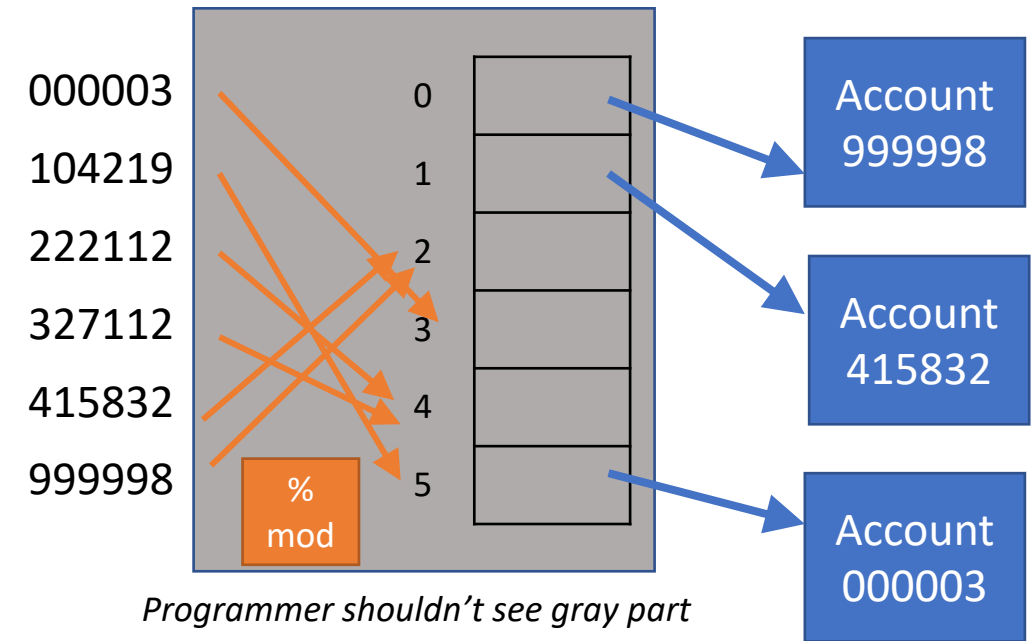
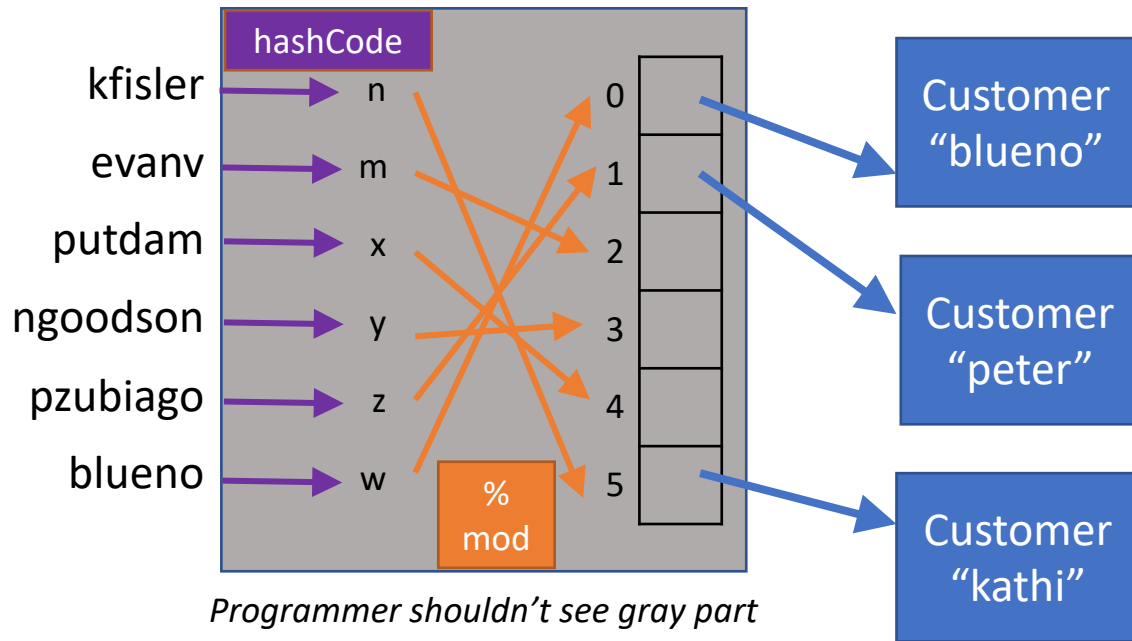


Implementing Hashtables

Remember this picture from the last class:



```
import java.util.HashMap;
```

```
private HashMap<Integer, Account> accounts = new HashMap<Integer, Account>();
private HashMap<String, Customer> customers = new HashMap<String, Customer>();
```

Today's goal: discuss how to implement a HashMap class for ourselves (finish for hwk!)

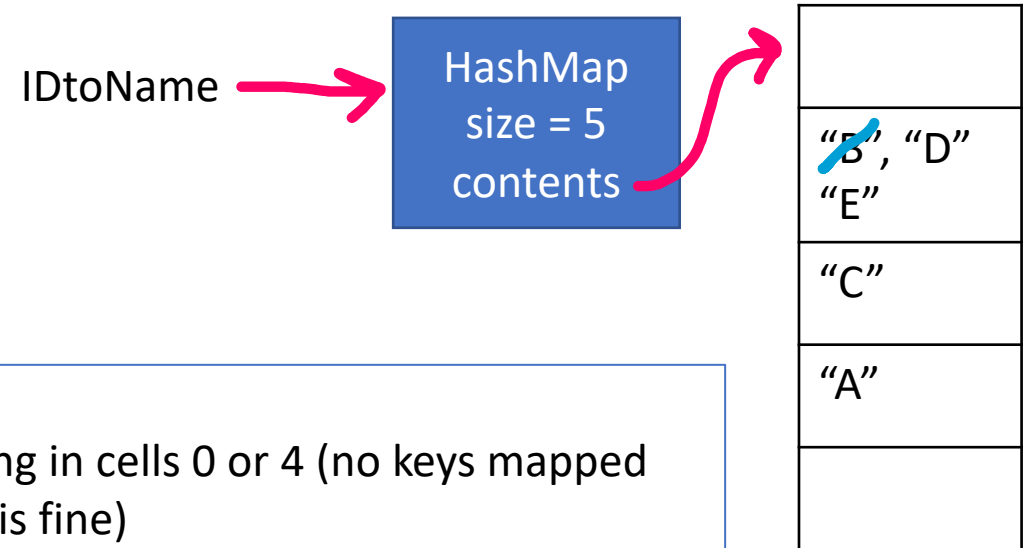
Memory Map – what's underneath? (take 1)

```
import java.util.HashMap;

HashMap<Integer,String> IDtoName;
IDtoName = new HashMap<>(5);
IDtoName.put(100003, "A");
IDtoName.put(100016, "B");
IDtoName.put(100007, "C");
IDtoName.put(100021, "D");
IDtoName.put(100016, "E");
```

we map keys to indices by taking key value modulo (remainder under division) the array size. % is the modulo operator in Java.

```
100003 % 5 = 3
100016 % 5 = 1
...
```



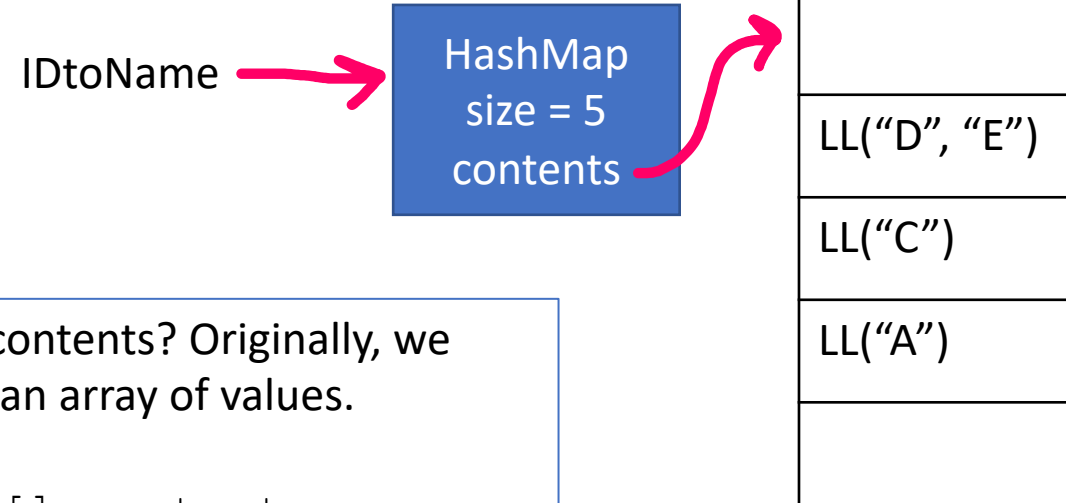
Note:

- there's nothing in cells 0 or 4 (no keys mapped there, which is fine)
- when we got a new value for key 100016 (E), it replaced the old value for 100016 (B)
- when we got a second key that mapped to index 1 (100021, after 100016), we had to store both values in the array

Memory Map – what's underneath? (take 1)

```
import java.util.HashMap;

HashMap<Integer,String> IDtoName;
IDtoName = new HashMap<>(5);
IDtoName.put(100003, "A");
IDtoName.put(100016, "B");
IDtoName.put(100007, "C");
IDtoName.put(100021, "D");
IDtoName.put(100016, "E");
```



What is the type of contents? Originally, we thought it would be an array of values.

```
String[] contents
```

Now it looks like the array has to hold multiple values. Try a list?

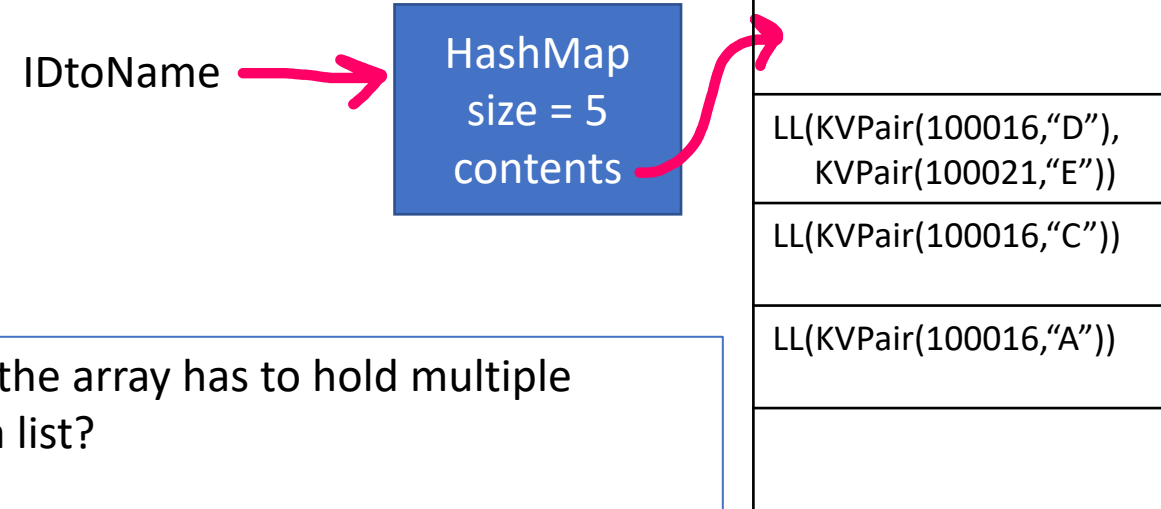
```
LinkedList<String>[] contents
```

Memory Map – what's underneath? (take 1)

```
import java.util.HashMap;

HashMap<Integer,String> IDtoName;
IDtoName = new HashMap<>(5);
IDtoName.put(100003, "A");
IDtoName.put(100016, "B");
IDtoName.put(100007, "C");
IDtoName.put(100021, "D");
IDtoName.put(100016, "E");
```

Internally, a hashtable
maps keys to lists of
key-value pairs



It looks like the array has to hold multiple values. Try a list?

```
LinkedList<String>[] contents
```

But if we are asked for the value associated with key 100016, how do we know whether "D" or "E" is the right value?

We have to store keys and their values in the list

```
LinkedList<KVPair<Integer,String>>[] contents
```