HashTables

What is a hashcode?

A hashcode is used to create a key for an item.

The hashcode may be the same as the value of the item or may involve the use of some elaborate formula in hopes of creating a unique key.

What is a hashcode?

```
Character c = new Character('a');
out.println(c.hashCode());

c = new Character('0');
out.println(c.hashCode());

c = new Character('A');
out.println(c.hashCode());
```

OUTPUT

```
97
48
65
```

What is a hashcode?

```
Integer num = 45;
out.println(num.hashCode());

num = new Integer(101);
out.println(num.hashCode());

String s = "a";
out.println(s.hashCode());
```

OUTPUT

```
45
101
97
```

Example HashCode Methods

```
public int hashCode( )
{
  return  x % someSize;
}

public int hashCode( )
{
  return (numVowels(s)+s.length)%aSize;
}
```

Open hashcode.java

What is a hash table?

A hash table stores values based on the hash code of each value.

```
Object[] hashTable = new Object[10];

Character c = new Character('1');
hashTable[c.hashCode()%10] = c;
```

What is a hash table?

```
Object[] hashTable = new Object[10];

Character c = new Character('1');
hashTable[c.hashCode()%10] = c;

Integer num = new Integer(113);
hashTable[num.hashCode()%10] = num;

String s = "e";
hashTable[s.hashCode()%10] = s;

for( Object thing : hashTable )
{
  System.out.println(thing);
}
```

OUTPUT

```
null
e
null
113
null
null
null
null
null
1
```

Open hashtableone.java

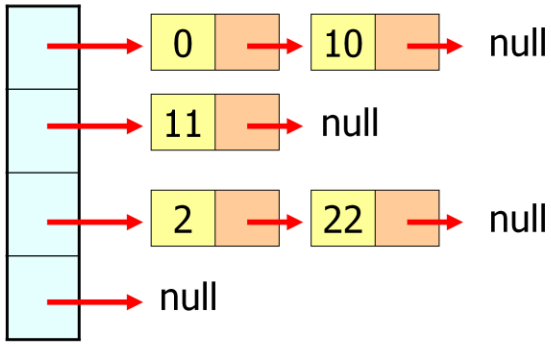
What is a hash table?

Most hash tables are built using an array of linked lists.

```
LinkedList[]   table;

table  = new LinkedList[10];
```

What is a hash table?



What is a hash table?

```
LinkedList[] hashTable = new LinkedList[5];

for( LinkedList list : hashTable )
{
  System.out.println(list);
}

for(int i = 0; i< hashTable.length; i++)
{
  hashTable[i] = new LinkedList();
}

for( LinkedList list : hashTable )
{
  System.out.println(list);
}
```

OUTPUT

```
null
null
null
null
null
[]
[]
[]
[]
[]
```

Open hashtabletwo.java

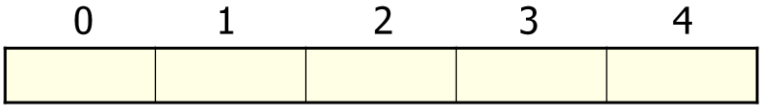
Open hashtablethree.java

Open hashtablefour.java

HashTable

HashSet and HashMap were both created around hash tables.

A hash table is a giant array. Each item is inserted into the array according to a hash formula.



MAPS

Key	Value
restroom	bano
cat	gato
boy	muchacho
house	casa
toad	sapo
water	agua

Hash tables are similar to maps.

Both structures store each key and the value associated with that key.

HashTable

Key	Value
0	100
1	31
2	22
3	93
4	64
5	5

In most hash tables, the key involves some manipulation of the hash code and the value is just the thing being stored.

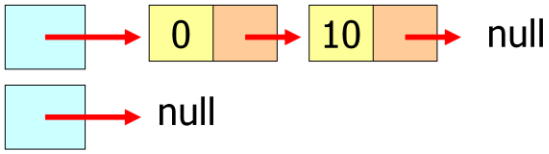
Collisions

Anytime you use a hash formula to generate index positions, you could end up with the same index position for several values.

Collisions are handled by putting all the values with the same hash code in a list and storing the list at the index position that matches the hash code.

Bucket Hashing/Chaining

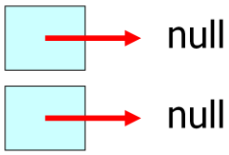
Using a bucket or a chain is a common way to describe putting all of the values with the same hash code in one list.



Bucket Hashing/Chaining

With only two buckets, this table would have lots of collisions and very long chains and be quite pointless.

```
LinkedList[]  table  = new LinkedList[2];
```



Bucket Hashing/Chaining

The larger the array, the less likely you are to have collisions. The table will be much more efficient as well.

```
LinkedList[]   table  = new LinkedList[100];

LinkedList[]   bigTable  = new LinkedList[1000];
```

Using ListNode

The LinkedList class works well to chain items together, but ListNode could also be used.

```
ListNode[]   table   = new ListNode[100];

ListNode[]   bigTable   = new ListNode[1000];
```

Open hashtablefive.java

HashTable
Big O Notation

Best Case / Average Case BigO of O(1)

Worst Case - BigO of O(N)

The Java HashSet and HashMap classes have a BigO of O(1). This is considered a generous BigO.

Start work on Lab 16