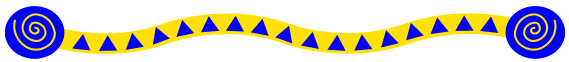
Arrays and Lists



Lab 14

What is a list?

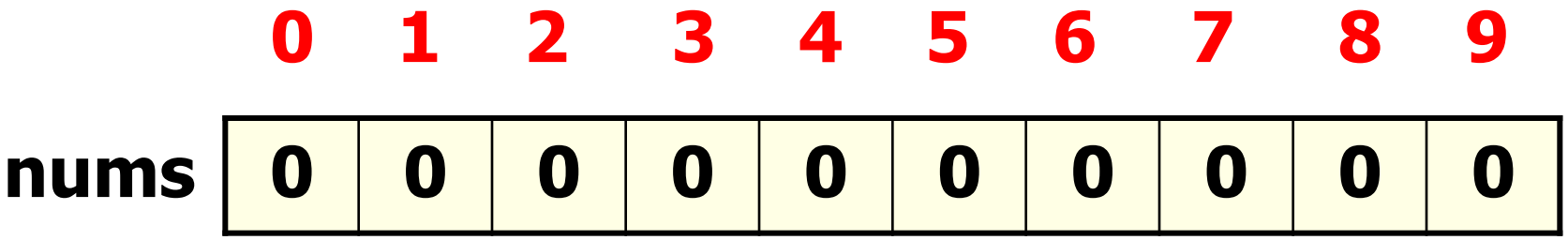
	Name	Time	Artist	Album	Ge
5	I Dare You to Move	4:08	Switchfoot	Learning to Breathe	
6	I've Been Everywhere	3:20	Johnny Cash	Unchained	
7	Brown Eyed Girl (Single Version)	3:05	Van Morrison	Super Hits	
8	Born to Be Wild	3:31	Steppenwolf	Steppenwolf: All Time Greatest	
9	Magic Carpet Ride	4:28	Steppenwolf	Steppenwolf: All Time Greatest	
10	Crazy (Single Version)	2:42	Patsy Cline	Patsy Cline's Greatest Hits (Ren	
11	Brick House	3:46	The Commodores	20th Century Masters - The Mill	
12	Cleveland Rocks	2:33	The Presidents of the...	Pure Frosting	
13	Chariots of Fire: Main Title Theme	3:32	Carl Davis & Royal Li...	Great Movie Themes	
14	Dueling Banjos (From "Deliverance")	3:11	The Hit Crew	Smash Hit Dramas Movie Theme	
15	Main Theme (From "Superman")	4:12	John Williams	The Music of John Williams - 40	
16	Main Theme (From "Superman")	4:12	John Williams	The Music of John Williams - 40	
17	I've Been Everywhere	3:20	Johnny Cash	Unchained	
18	Born to Be Wild	3:31	Steppenwolf	Steppenwolf: All Time Greatest	

What is an Array?

What is an array?

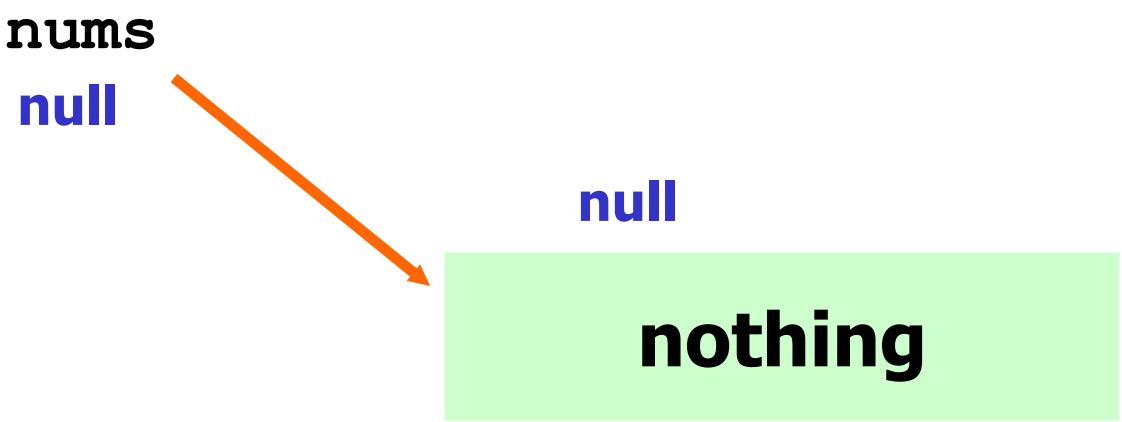
An array is a group of items all of the same type which are accessed through a single identifier.

```
int[] nums = new int[10];
```



Array References

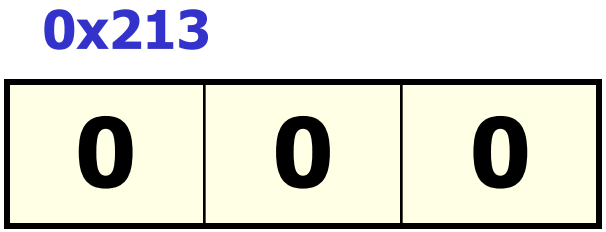
```
int[] nums;
```



nums is a reference to an integer array.

Array Instantiation

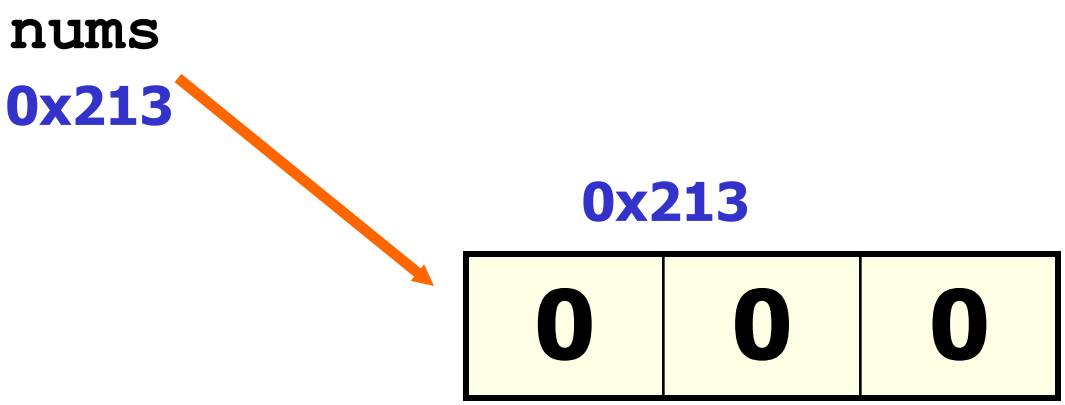
new int[3];



arrays are Objects.

Arrays

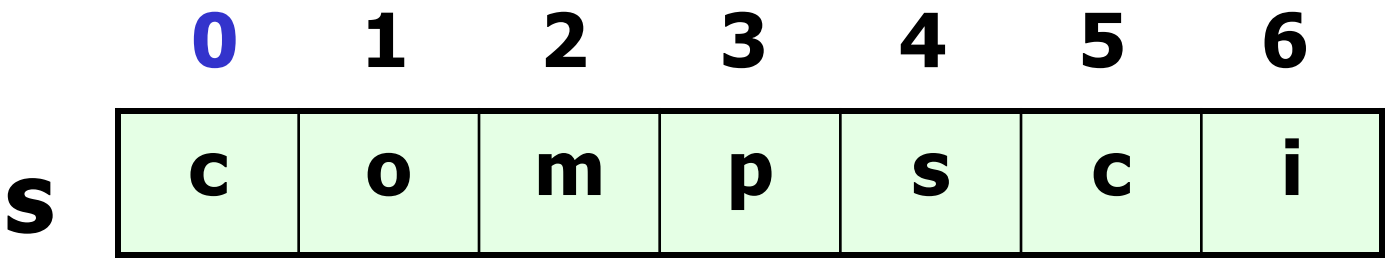
```
int[] nums = new int[3];
```



nums is a reference to an integer array.

Strings are arrays

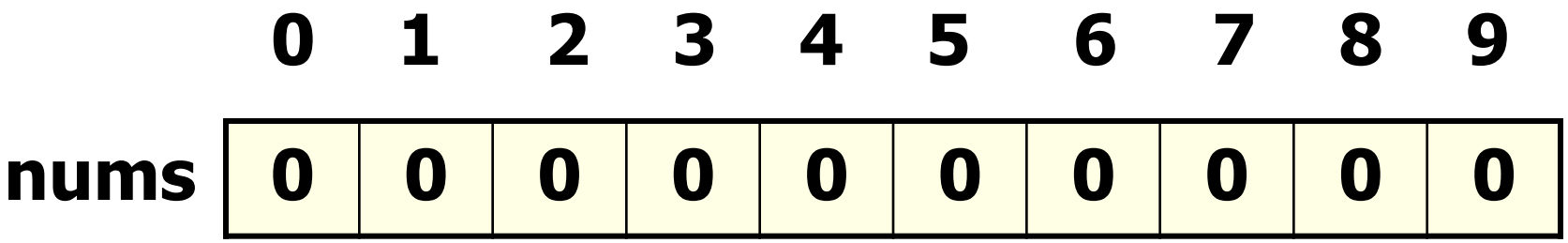
```
String s = "compsci";      //Strings are arrays
```



The first index position in a String is 0.
A String is an array of characters.

Arrays

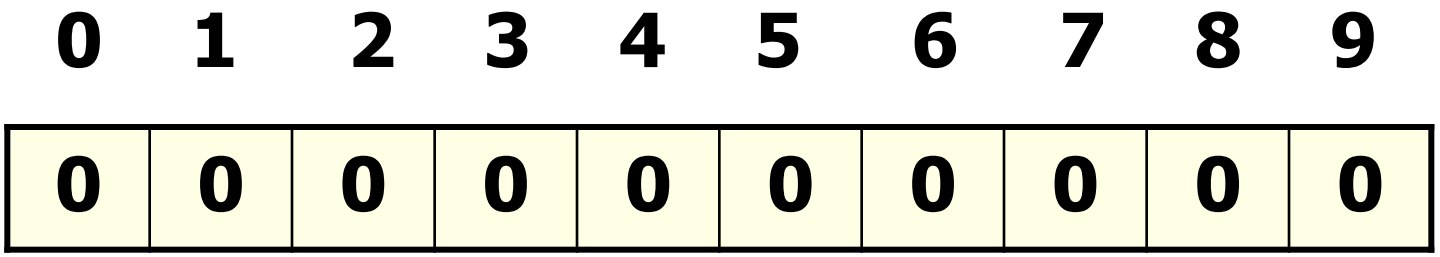
```
int[] nums = new int[10];        //Java int array
```



Arrays are filled with 0 values when instantiated. The exact value of each spot in the array depends on the specified type for the array.

Arrays

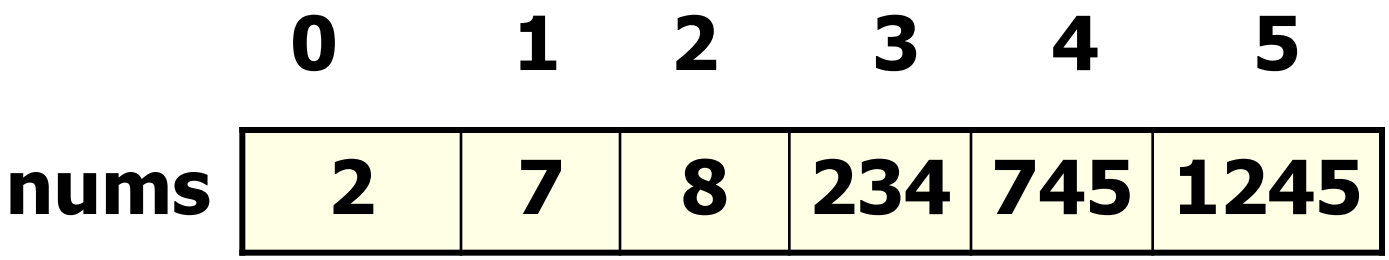
```
new int[10];      //Java int array
```



Once an array object has been instantiated, the size many never change. To increase or decrease the size, a new array would need to be instantiated and all old value copied.

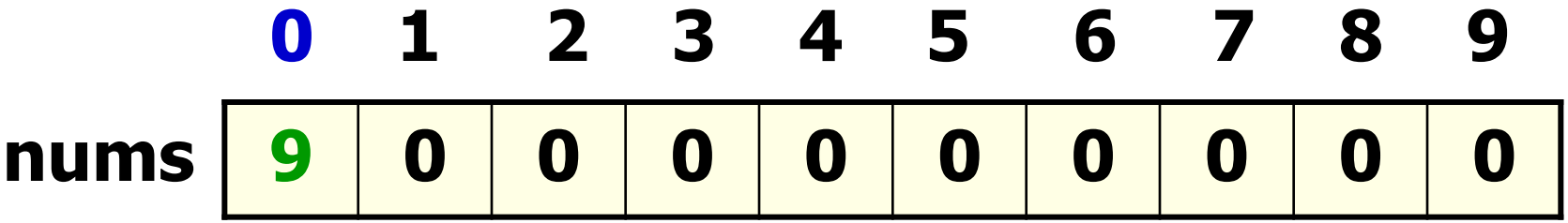
Arrays

```
int[] nums = {2,7,8,234,745,1245};
```



An array can be initialized with values.

Indexes

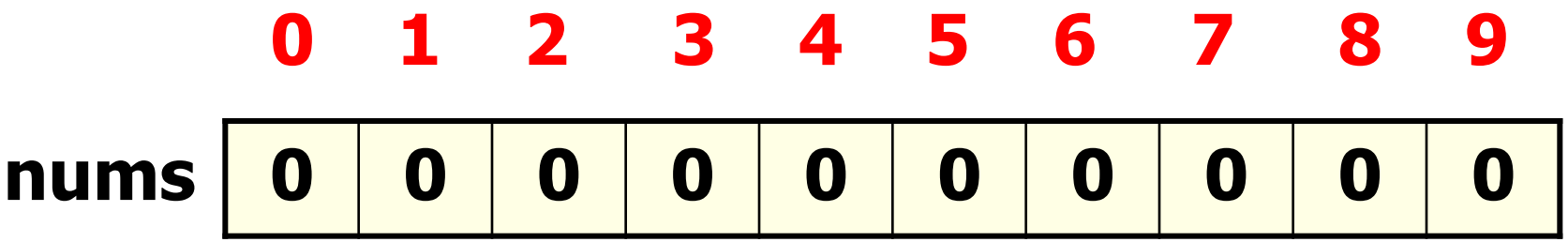


The [spot/index] indicates which value in the array is being manipulated.

nums[0] = 9;
The 0 spot is being set to 9.

Indexes

Java indexes must always be *integers* and the first index will always be 0.



arrayinit.java

Printing Array Values

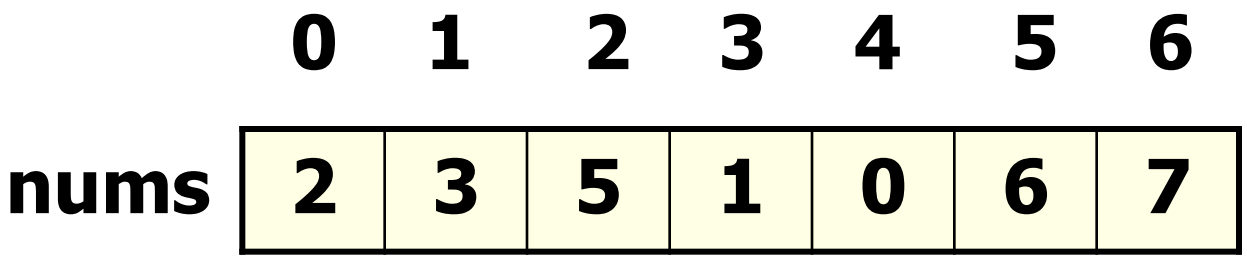
Printing Array Values

```
int[] nums = {2,3,5,1,0,6,7};

out.println(nums[0]);
out.println(nums[2]);
out.println(nums[5]);
```

OUTPUT

```
2
5
6
```



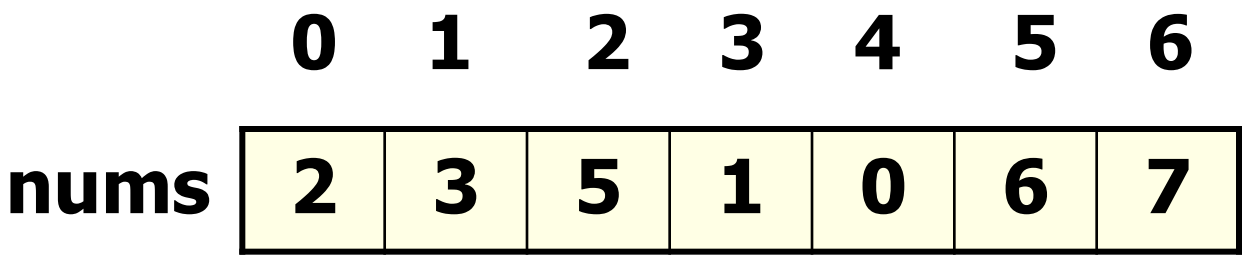
Printing Array Values

```
int[] nums = {2,3,5,1,0,6,7};

out.println( nums[ 1 + 3 ] );
out.println( nums[ 7 / 2 ] );
out.println( nums[ 6 ] );
```

OUTPUT

```
0
1
7
```



open arrayprintone.java arrayprinttwo.java

Setting Array Spots

Setting array spots

```
int[] nums = new int[10];

nums[0] = 231;
nums[4] = 756;
nums[2] = 123;

out.println(nums[0]);
out.println(nums[1]);
out.println(nums[4]);
out.println(nums[4/2]);
```

OUTPUT

```
231
0
756
123
```

Setting array spots

```
double[] nums = new double[10];

nums[0] = 10.5;
nums[3] = 98.6;
nums[2] = 77.5;

out.println(nums[0]);
out.println(nums[3]);
out.println(nums[7]);
```

OUTPUT

```
10.5
98.6
0.0
```

open arraysetone.java arraysettwo.java

Accessing Arrays with Loops

Accessing Arrays with Loops

```
int[] nums = {3,2,5,1,0,6};
for(int spot=0; spot<nums.length; spot++)
{
    out.println(nums[spot]);
}
```

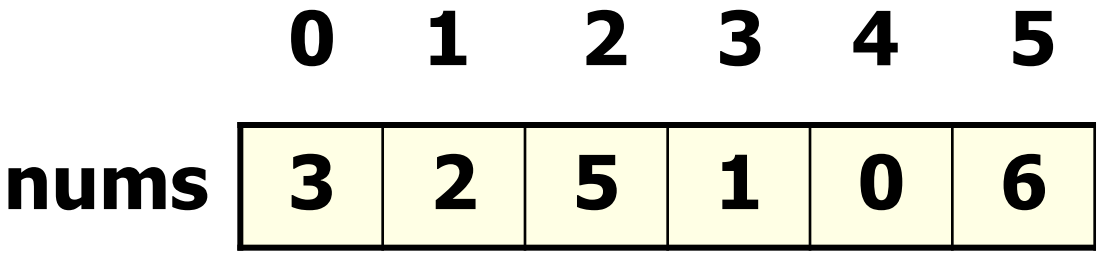
length returns the # of elements/items/spots in the array!!!

OUTPUT

```
3
2
5
1
0
6
```

Accessing Arrays with Loops

```
int[] nums = {3,2,5,1,0,6};
for(int  item  :  nums)
{
    out.println(item);
}
```

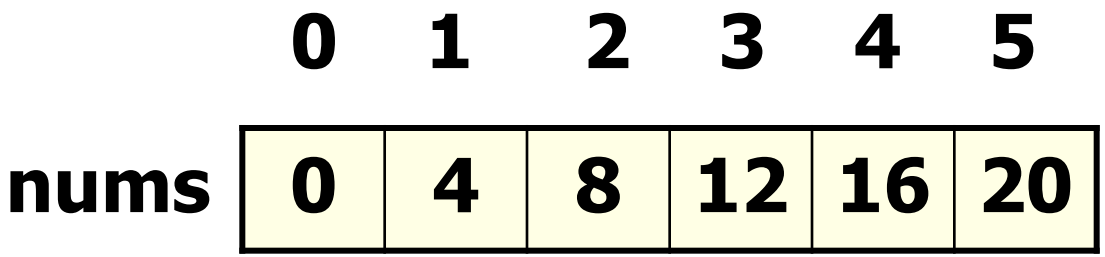


OUTPUT

```
3
2
5
1
0
6
```

Accessing Arrays with Loops

```
int[] nums = new int[6];
for(int spot=0; spot<nums.length; spot++)
{
   nums[spot] = spot*4;
}
```



open arrayloopone.java arraylooptwo.java

Counting Array Values

Counting Array Values

In order to count the number of occurrences of a particular value, you must use a loop to access all items in the array. You must also include an if statement to check for the search val and a variable with which to count each of the matches.

```
for( loop through all items )
{
   if( item matches search value )
       increase the count by 1
}
```

Counting Array Values

```
int[] nums =   //some set of numbers
int count = 0;

for( int  item  :  nums )
{
    if ( item matches provided value )
       count = count + 1;
}
```

arraycount.java

Deleting Array Values

Deleting Array Values

Once instantiated, the size of an array can never change.

```
int[] nums = {1,7,8,7,4,3,7};
```

Deleting Array Values

To delete values, a new array must be instantiated.
Old values will be copied to the new array.

```
int[] nums = {1,7,8,7,4,3,7};
```

Deleting Array Values

If all 7s are to removed from the array below, what must be done?

```
int[] nums = {1,7,8,7,4,3,7};
```

Deleting Array Values

```
int[] nums = {1,7,8,7,4,3,7};
```

To delete all 7s

Count the 7s
Make a new array
Size it to non 7s count
Copy all non 7s to new array

arraydelete.java

Arrays as Instance Variables

Instance Variables

```
public class Array
{
   private int[]  nums;      //has the value null

   public Array(){
      nums = new int[10];    //sizes the array
   }

   //other methods not shown
}
```

arrayinstancevars.java

toString()

```
public class Array
{
  //instance vars and other methods not shown

  public String toString()
  {
     String output= "";
     for(int spot=0; spot<nums.length; spot++)
     {
        output=output+nums[spot]+" ";
     }
     return output;
  }
}
```

toString()

```
public class Array
{
  //instance vars and other methods not shown

  public String toString()
  {
     String output= "";
     for(  int val  :  nums  )
     {
        output = output + val + " ";
     }
     return output;
  }
}
```

arrayinstancevarstwo.java

InstanceVarsTwo

```
String list = "7 6 3 4 9 1 3 5";
int[] nums = new int[8];

Scanner chopper = new Scanner(list);
int spot=0;

while(chopper.hasNextInt())
{
  nums[spot++]=chopper.nextInt();
}
```

Sorting Arrays

sort

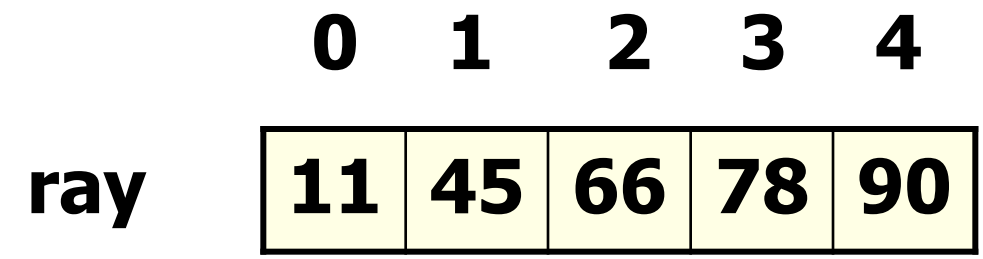
```
int nums[] = {45,78,90,66,11};

Arrays.sort(nums);

for(int item : nums)
   out.println(item);
```

OUTPUT

```
11
45
66
78
90
```



open sort.java

Start work on Lab 14