A+ Computer Science

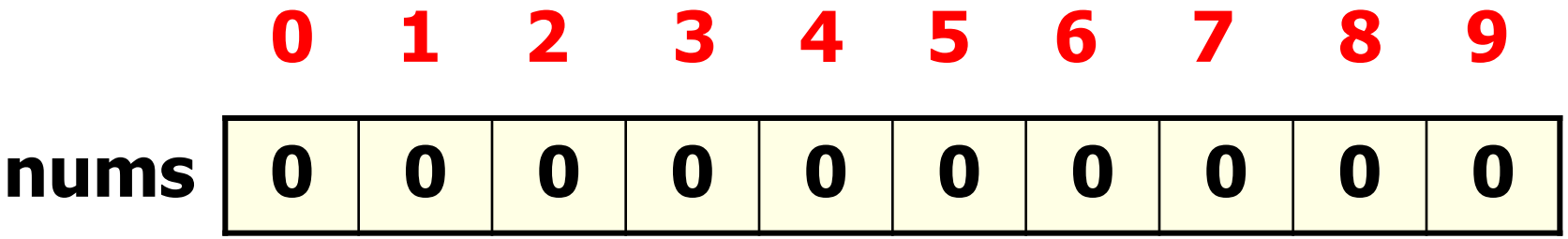
JAVA ARRAYS

What is an array?

What is an array?

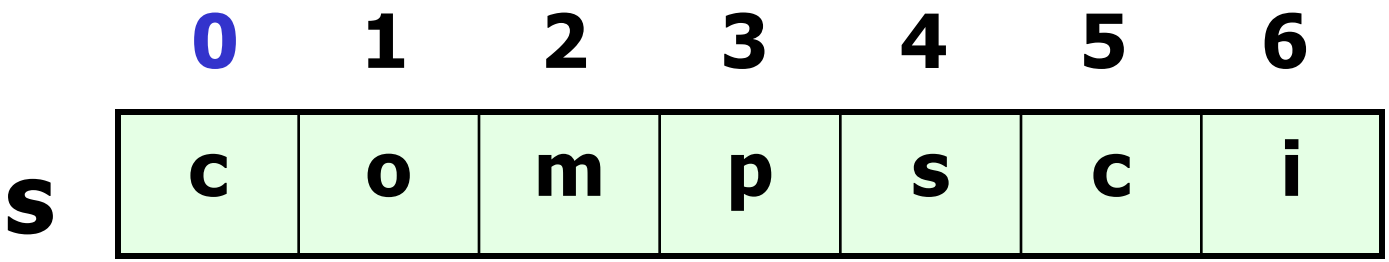
An array is a group of items all of the same type which are accessed through a single identifier.

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



What is an array?

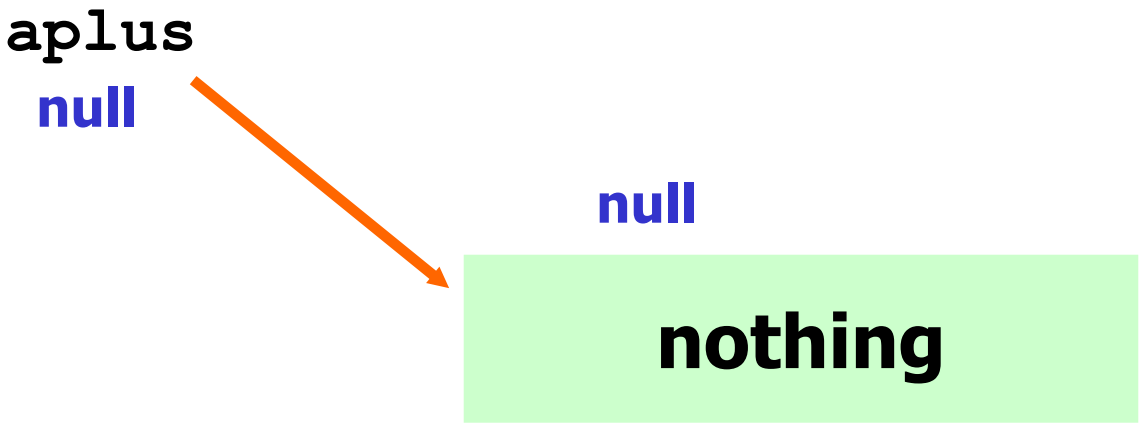
```
String s = "compsci";      //Strings are arrays
```



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

What is a reference?

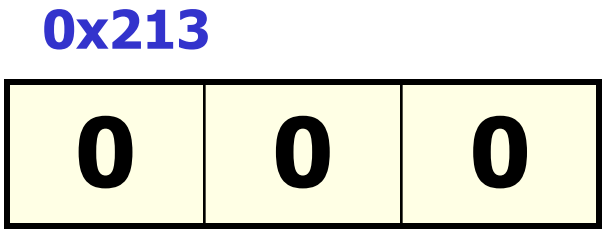
```
int[]  aplus;
```



aplus is a reference to an integer array.

Array Instantiation

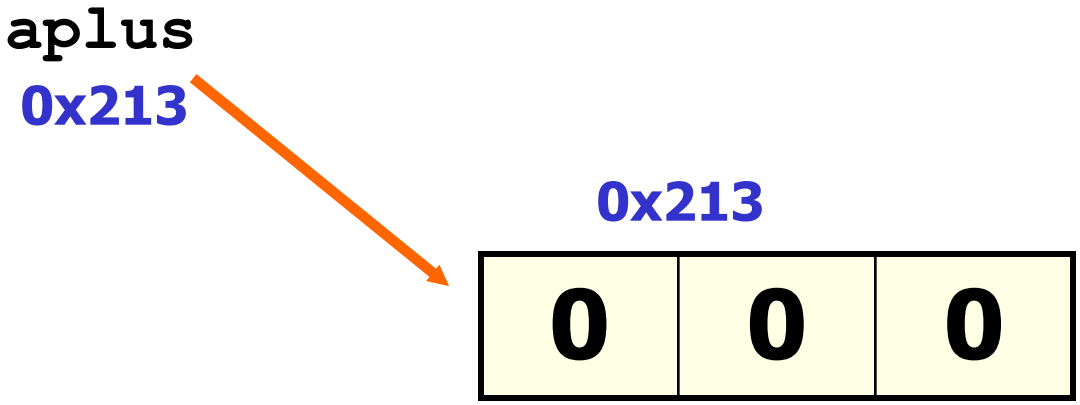
new int[3];



arrays are Objects.

What is an array?

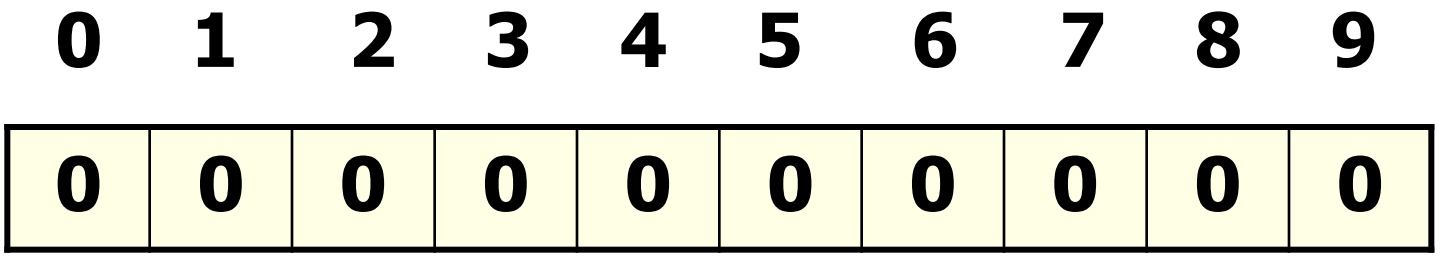
```
int[] aplus = new int[3];
```



aplus is a reference to an integer array.

What is an array?

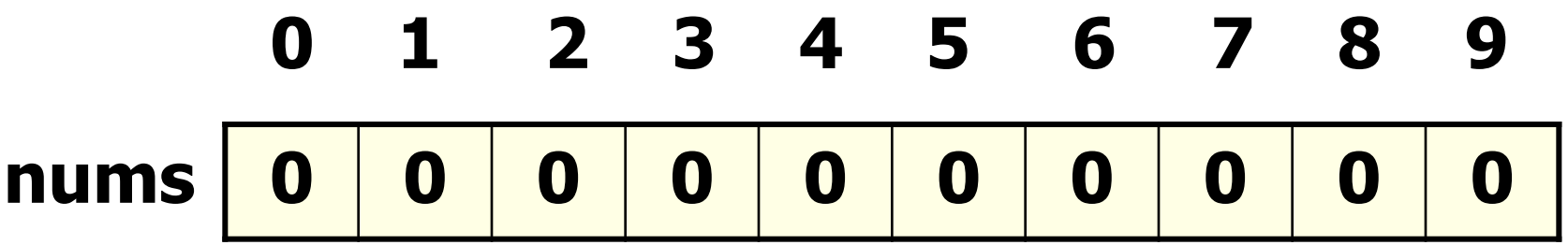
```
new int[10];     //Java int array
```



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

What is an array?

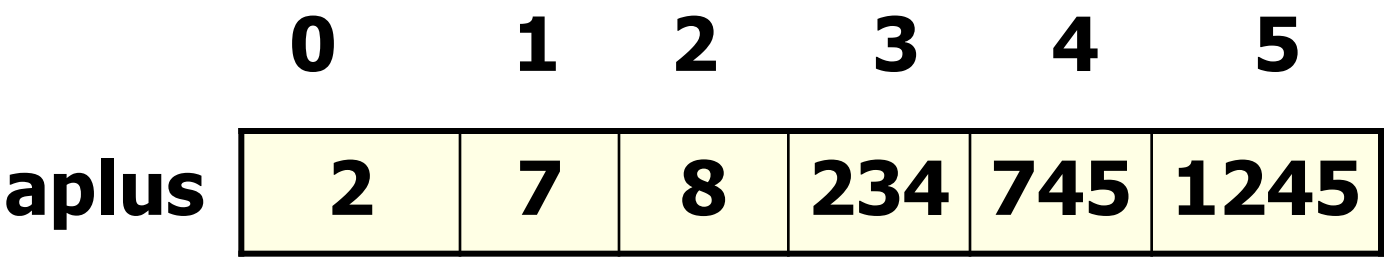
```
int[] aplus = new int[10];     //Java int array
```



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.

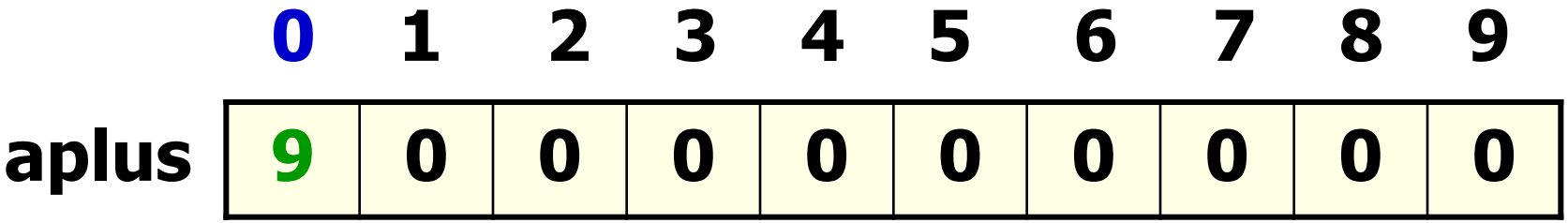
What is an array?

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



An array can be initialized with values.

What is an array?

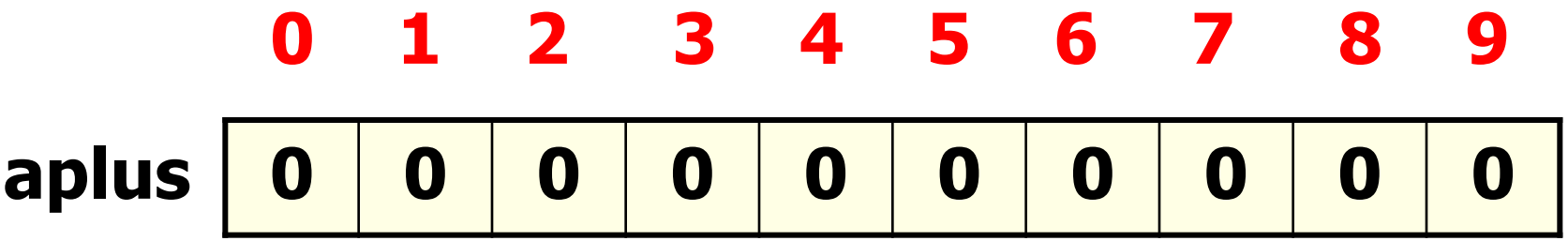


The [spot/index] indicates which value in the array is being manipulated.

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

What is an array?

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



arrayinit.java

Printing Array Values

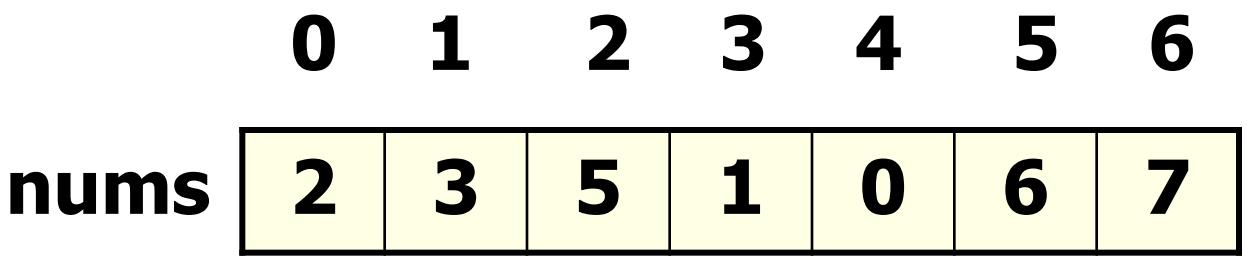
Printing Array Values

```
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
```



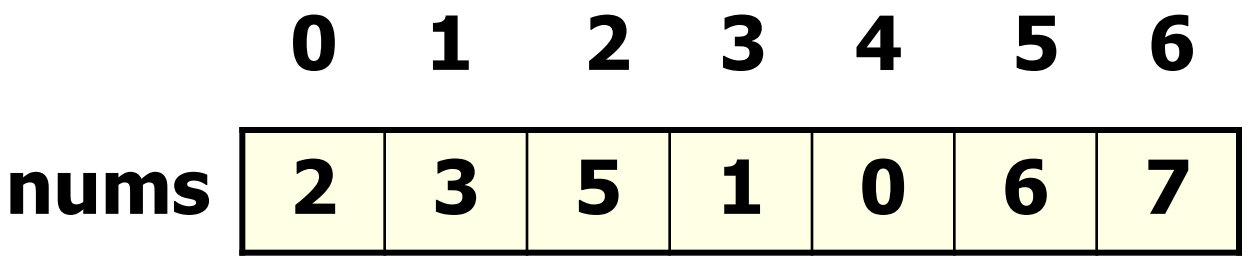
Printing Array Values

```
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
```



arrayprintone.java
arrayprinttwo.java

Setting Array Spots

Setting Array Values

```
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 Values

```
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
```

Setting Array Values

```
String[] words = new String[10];

words[0] = "dog";
words[3] = "cat";
words[2] = "pig";

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

OUTPUT

```
dog
cat
null
```

arraysetone.java
arraysettwo.java

Accessing Arrays with Ifs

Accessing Array Values

```
int[] nums = {1,2,3,4,5,6,7};

if( nums[1] == nums[2] )
  out.println( "aplus" );
else
  out.println( "comp" );
```

OUTPUT

comp

Accessing Array Values

```
int[] nums = {1,2,3,4,5,6,2};

int last = nums.length-1;
if( nums[1] == nums[last] )
  out.println( "aplus" );
else
  out.println( "comp" );
```

OUTPUT

aplus

Accessing Array Values

```
int[] nums = {1,2,3,4,5,6,2};

int last = nums.length;
if( nums[last] == 7 )
  out.println( "aplus" );
else
  out.println( "comp" );
```

OUTPUT

arrayindex outof bounds exception

arrayifsone.java
arrayifstwo.java

Traversing Arrays with Loops

Traversing Arrays

```
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

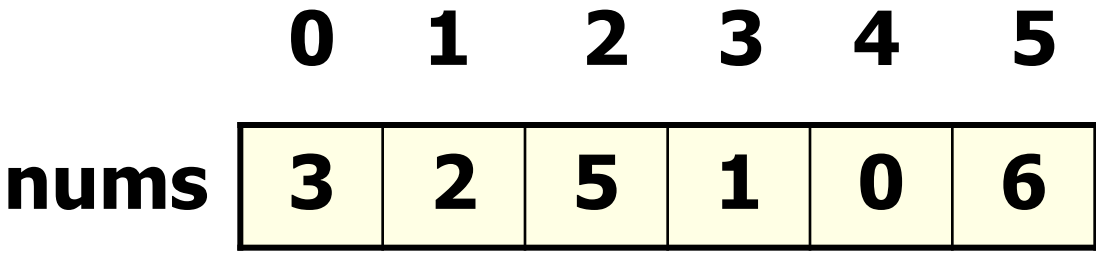
```
3
2
5
1
0
6
```

Traversing Arrays

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

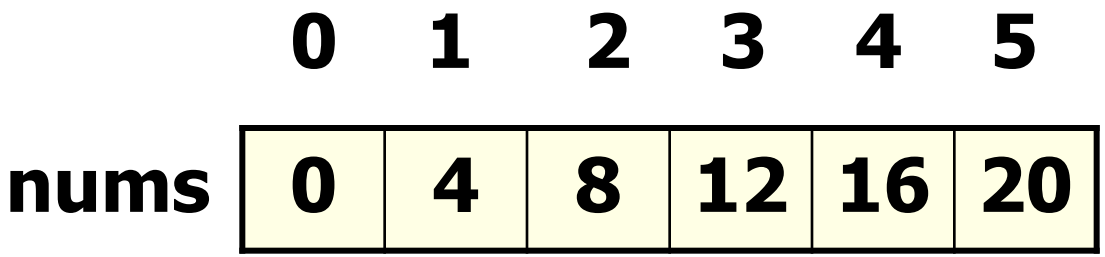
OUTPUT

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



Traversing Arrays

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

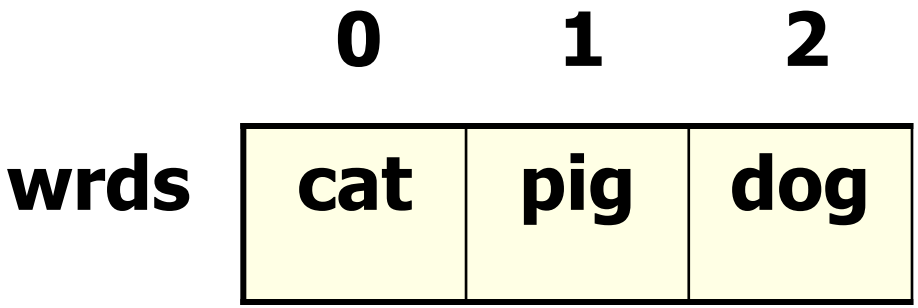


Traversing Arrays

```
String[] wrds = {"cat","pig","dog"};
for(String  item  :  wrds)
{
    out.println(item);
}
```

OUTPUT

cat
pig
dog



Traversing Arrays

```
int[] nums = {2,2,3,3,4,5,5};
for(int x=0; x<=nums.length; x++)
{
   nums[spot] = x*2;
}
```

This for loop runs one time too many as the loop condition has x<=nums.length instead of having x < nums.length. This results in an ArrayIndexOutOfBoundsException.

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 specified value and a variable with which to count each of the variable's occurrences.

Counting Array Values

```
loop through all array items

    if current item == search value

        increase the count by 1
```

Counting Array Values

```
//assume nums is an array with values

int count = 0;
for( int  i = 0; i < nums.length; i++ )
{
    if ( nums[ i ] matches some value )
       count = count + 1;
}

//return or print count
```

Counting Array Values

```
//assume nums is an array with values

int count = 0;
for( int  item  :  nums )
{
    if ( item matches some value )
       count = count + 1;
}

//return or print count
```

arraycount.java

Work on Programs!

Crank Some Code!

Arrays as Instance Variables

Array 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
}
```

arrayivars.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;
  }
}
```

arrayivarstwo.java

Array Parameters

Passing by Value

```
public static void changeOne(int[] ray)
{
  ray[0] = 0;
  ray[1] = 1;
}
```

OUTPUT

```
[5, 4, 3, 2, 1]
[0, 1, 3, 2, 1]
```

```
//test code
int[] nums = {5,4,3,2,1};
out.println(Arrays.toString(nums));
changeOne(nums);
out.println(Arrays.toString(nums));
```

Passing by Value

```
public static void changeOne(int[] ray)
{
  ray[0] = 0;
  ray[1] = 1;
}
```

Changing the values inside the array referred to by ray is a lasting change.
A copy of the original reference was passed to method changeOne, but that reference was never modified.

Passing by Value

```
public static void changeTwo(int[] ray)
{
   ray = new int[5];
   ray[0]=2;
   out.println(Arrays.toString(ray));
}

//test code
int[] nums = {5,4,3,2,1};
changeTwo(nums);
out.println(Arrays.toString(nums));
```

OUTPUT

```
[2, 0, 0, 0, 0]
[5, 4, 3, 2, 1]
```

Passing by Value

```
public static void changeTwo(int[] ray)
{
  ray = new int[5];
  ray[0]=2;
}
```

Referring ray to a new array has local effect, but does not affect the original reference.
A copy of the original reference was passed to method changeTwo.

arrayparams.java

Just For Fun

The Arrays Class

sort

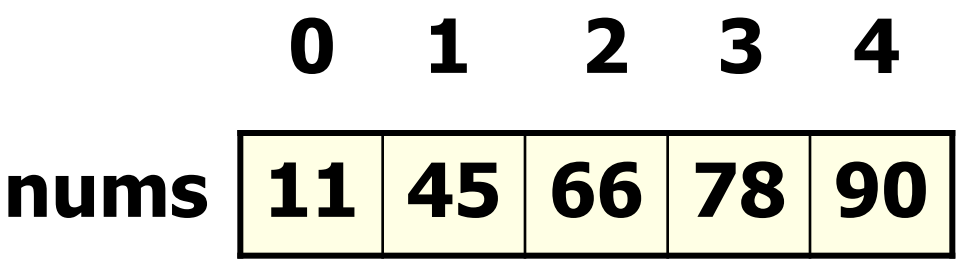
```
int nums[] = {45,78,90,66,11};

Arrays.sort(nums);

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

OUTPUT

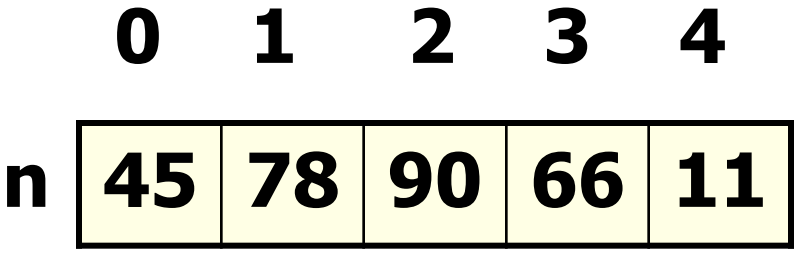
```
11
45
66
78
90
```



toString( )

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

System.out.println( Arrays.toString(n));
```



OUTPUT

```
[45, 78, 90, 66, 11]
```

arrays_class.java

Work on Programs!

Crank Some Code!

A+ Computer Science

JAVA ARRAYS