

Oct 16-7:52 AM

Quick Review: Arrays

Creating Arrays: `int[] arrayName = { element, element, ... }`

Nov 20-12:25 PM

Quick Review: Arrays

Creating Arrays: `int[] arrayName = { element, element, ... }`

Array Elements: Starts with index 0, then 1, 2, ...

Quick Review: Arrays

Creating Arrays: `int[] arrayName = { element, element, ... }`

Array Elements: Starts with index 0, then 1, 2, ...

For Loops in Arrays: `for(int i=0 ; i<arrayName.length ; i++)``for(int i=1 ; i<=arrayName.length ; i++)`***** Same ... but not typically used in arrays!!! *****

Nov 20-12:25 PM

Nov 20-12:25 PM

Quick Review: Arrays

Creating Arrays: `int[] arrayName = { element, element, ... }`

Array Elements: Starts with index 0, then 1, 2, ...

For Loops in Arrays: `for(int i=0 ; i<arrayName.length ; i++)`
`for(int i=1 ; i<=arrayName.length ; i++)`Enhanced for-loops: `for(int number : arrayName)`

Quick Review: Arrays

Creating Arrays: `int[] arrayName = { element, element, ... }`

Array Elements: Starts with index 0, then 1, 2, ...

For Loops in Arrays: `for(int i=0 ; i<arrayName.length ; i++)`
`for(int i=1 ; i<=arrayName.length ; i++)`Enhanced for-loops: `for(int number : arrayName)`Array Lengths: `arrayName.length`

Nov 20-12:25 PM

Nov 20-12:25 PM

Unit 06 Day 01 - Review and Swapping Array Elements.notebook

February 24, 2016

A very basic Array ...

```
int[ ] arrayName = { 2, 1, 5 }
```

We've been told the numbers need to be in order ...

*** Swap the 1 and the 2 ***

We need to temporarily
store this value before
it gets over-written!

```
arrayName[0] = arrayName[1];  
arrayName[1] = arrayName[0];
```

DOES NOT WORK!
DISCUSS

Nov 20-12:25 PM

A very basic Array ...

```
int[ ] arrayName = { 2, 1, 5 }
```

We've been told the numbers need to be in order ...

*** Swap the 1 and the 2 ***

```
arrayName[0] = arrayName[1];  
arrayName[1] = arrayName[0];
```

DOES NOT WORK!
DISCUSS

Nov 20-12:25 PM

A very basic Array ...

```
int[ ] arrayName = { 2, 1, 5 }
```

We've been told the numbers need to be in order ...

*** Swap the 1 and the 2 ***

We need to temporarily
store this value before
it gets over-written!

```
arrayName[0] = arrayName[1];  
arrayName[1] = arrayName[0];
```

DOES NOT WORK!
DISCUSS

Nov 20-12:25 PM

A very basic Array ...

```
int[ ] arrayName = { 2, 1, 5 }
```

We've been told the numbers need to be in order ...

*** Swap the 1 and the 2 ***

```
int temp = arrayName[0]; //store arrayName[0]  
arrayName[0] = arrayName[1]; //overwrite arrayName[0]  
arrayName[1] = temp; //change arrayName[1]
```

Nov 20-12:25 PM

```
int[ ] arrayName = { 2, 1, 5 }
```

```
int temp = arrayName[0]; //store arrayName[0]  
arrayName[0] = arrayName[1]; //overwrite arrayName[0]  
arrayName[1] = temp; //change arrayName[1]
```

Now arrayName = { 1, 2, 5 }

Nov 20-12:25 PM

```
int[ ] arrayName = { 2, 1, 5 }
```

```
int temp = arrayName[0]; //store arrayName[0]  
arrayName[0] = arrayName[1]; //overwrite arrayName[0]  
arrayName[1] = temp; //change arrayName[1]
```

Now arrayName = { 1, 2, 5 }

```
for(int number : arrayName)  
    System.out.print(number + " ");
```

Nov 20-12:25 PM

Moral of the Story (Want to swap Array elements? ...) ...

```
int temp = arrayName[0];           //store an element somewhere else
arrayName[0] = arrayName[1];       //overwrite the element
arrayName[1] = temp;              //assign 2nd element the stored value
```

Nov 20-12:25 PM

What if it isn't about swapping?
What if you want to completely reverse order?

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

*** Completely reverse to 1, 2, 3, 4, 5, 6, 7 ***

Nov 20-12:25 PM

Analysis ...

NOTE: arrayOfHope.length = 7

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



```
arrayOfHope[0] = arrayOfHope[6]
arrayOfHope[6] = arrayOfHope[0]
```

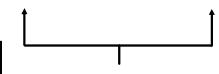
*** This DOES NOT work ***

Nov 20-12:25 PM

Analysis ...

NOTE: arrayOfHope.length = 7

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



* Remember, need a temporary storage spot.

```
arrayOfHope[0] = arrayOfHope[6]
arrayOfHope[6] = arrayOfHope[0]
```

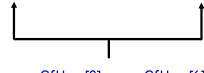
*** This DOES NOT work ***

Nov 20-12:25 PM

Analysis ...

NOTE: arrayOfHope.length = 7

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



* Remember, need a temporary storage spot.
* In this problem, every element gets swapped!

```
arrayOfHope[0] = arrayOfHope[6]
arrayOfHope[6] = arrayOfHope[0]
```

*** This DOES NOT work ***

Nov 20-12:25 PM

Analysis ...

NOTE: arrayOfHope.length = 7

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



* Remember, need a temporary storage spot.
* In this problem, every element gets swapped!

* We'll store all elements with a temp Array!

```
arrayOfHope[0] = arrayOfHope[6]
arrayOfHope[6] = arrayOfHope[0]
```

*** This DOES NOT work ***

Nov 20-12:25 PM

Analysis ...

NOTE: arrayOfHope.length = 7
 * We'll store all elements with a temp Array!

```
int[ ] arrayOfHope = { 7, 6, 5, 4, 3, 2, 1 };
int[ ] tempStorage = new int[arrayOfHope.length];
```

- Remember:**
1. If we don't fill an array we must declare its length.
 2. Should be the same length as arrayOfHope
 3. After creating the temporary Array, we need to fill it...

Nov 20-12:25 PM

Analysis ...

NOTE: arrayOfHope.length = 7
 * We'll store all elements with a temp Array!

```
int[ ] arrayOfHope = { 7, 6, 5, 4, 3, 2, 1 };
int[ ] tempStorage = new int[arrayOfHope.length]; //must declare length

for(int i=0 ; i<arrayOfHope.length ; i++)
  tempStorage[i] = arrayOfHope[i];
```

- Remember:**
1. This for-loop runs from index 0 to index 6 (good!)
 2. It makes a copy of arrayOfHope[] = tempStorage[]
 3. Once copied, tempStorage[] needs to be dumped into arrayOfHope[] but in the reverse...

Nov 20-12:25 PM

Here's how we do the swap ...

```
for(int i=0 ; i<arrayOfHope.length ; i++)
  arrayOfHope[i] = tempStorage[(arrayOfHope.length-1) - i]

↓

arrayOfHope[0] = tempStorage[(7-1) - 0]
↑   ↓
start at      This will allow us to
index 6        decrement: index 6, 5, 4, ...
```

Nov 20-12:25 PM

Here's how we do the swap ...

```
for(int i=0 ; i<arrayOfHope.length ; i++)
  arrayOfHope[i] = tempStorage[(arrayOfHope.length-1) - i]

↓

arrayOfHope[0] = tempStorage[(7-1) - 0] = tempStorage[6]
```

Nov 20-12:25 PM

Here's how we do the swap ...

```
for(int i=0 ; i<arrayOfHope.length ; i++)
  arrayOfHope[i] = tempStorage[(arrayOfHope.length-1) - i]

↓

arrayOfHope[0] = tempStorage[(7-1) - 0] = tempStorage[6]
arrayOfHope[1] = tempStorage[(7-1) - 1] = tempStorage[5]
```

Nov 20-12:25 PM

Here's how we do the swap ...

```
for(int i=0 ; i<arrayOfHope.length ; i++)
  arrayOfHope[i] = tempStorage[(arrayOfHope.length-1) - i]

↓

arrayOfHope[0] = tempStorage[(7-1) - 0] = tempStorage[6]
arrayOfHope[1] = tempStorage[(7-1) - 1] = tempStorage[5]
arrayOfHope[2] = tempStorage[(7-1) - 2] = tempStorage[4]
```

Nov 20-12:25 PM

Here's how we do the swap ...

```
for(int i=0 ; i<arrayOfHope.length ; i++)
    arrayOfHope[i] = tempStorage[(arrayOfHope.length-1) - i]

↓

arrayOfHope[0] = tempStorage[(7-1) - 0]] = tempStorage[6]
arrayOfHope[1] = tempStorage[(7-1) - 1]] = tempStorage[5]
arrayOfHope[2] = tempStorage[(7-1) - 2]] = tempStorage[4]
arrayOfHope[3] = tempStorage[(7-1) - 3]] = tempStorage[3]
...
arrayOfHope[6] = tempStorage[(7-1) - 6]] = tempStorage[0]
```

Nov 20-12:25 PM

Analysis ...

NOTE: `arrayOfHope.length = 7`
 * We'll store all elements with a temp Array!

```
int[] arrayOfHope = { 7, 6, 5, 4, 3, 2, 1 }
int[] tempStorage = new int[arrayOfHope.length];           //must declare length
for(int i=0 ; i<arrayOfHope.length ; i++)
    tempStorage[i] = arrayOfHope[i];
for(int i=0 ; i<arrayOfHope.length ; i++)
    arrayOfHope[i] = tempStorage[(arrayOfHope.length-1) - i]] //swap
```

- Remember:**
1. One array counts up while the other counts down
 2. One array follows for-counter, other uses length (end)
 3. Once swapped, we should print to verify ...

Nov 20-12:25 PM

Analysis ...

NOTE: `arrayOfHope.length = 7`
 * We'll store all elements with a temp Array!

```
int[] arrayOfHope = { 7, 6, 5, 4, 3, 2, 1 }
int[] tempStorage = new int[arrayOfHope.length];           //must declare length
for(int i=0 ; i<arrayOfHope.length ; i++)                //dump them in!
    tempStorage[i] = arrayOfHope[i];
for(int i=0 ; i<arrayOfHope.length ; i++)
    arrayOfHope[i] = tempStorage[(arrayOfHope.length-1) - i]] //swap-fill

for(int number : arrayOfHope)                            //enhanced for-loop is great!
    System.out.print(number+ " ");                      //used to access all elements
```

Nov 20-12:25 PM

Things to do ...

1. Work on Unit 06 WS01
2. Reminder - The Final is Near!

Oct 16-9:12 AM