1. Log in
2. Working in Unit 3
   - 1-Dimensional Arrays
   - 2-Dimensional Arrays
   - Array Lists

Oct 22-4:28 PM

More with the 1-D Array!

A basic class to count how many even numbers are in an array.

```
int counter=0;
int[ ] numbers = {2,4,5,6,7,8,9};

for(int i=0;i<numbers.length;i++)
   if(numbers[i]%2==0 && numbers[i]!=0)
      counter++;
System.out.println("There were "+counter+" even #'s in your array");
```

Sep 12-7:31 PM

Introducing the for-each loop ...
Also called the enhanced for-loop ...

A basic class to count how many even numbers are in an array.

```
int counter=0;
for(int x : numbers)          //for each integer 'x' in the array numbers
   if(x%2==0 && x!=0)
      counter++;
System.out.println("There were "+counter+" even #'s in your array");
```

*** The for-each loop is used when you need to look ***
*** at ALL elements of an array ... you can't stop it! ***
*** IT IS NOT USED FOR EDITING AN ARRAY!!! ***

*** We will now only be using for-each in 1D arrays ***

Sep 12-7:31 PM

Using a 1-D Array to create a deck of cards ...

Writing code ...

```
String[] deckOfCards = {"A of Diamonds","2 of Diamonds",
                        "3 of Diamonds","4 of Diamonds","5 of Diamonds",
                        "6 of Diamonds","7 of Diamonds","8 of Diamonds",
                        "9 of Diamonds","10 of Diamonds","J of Diamonds",
                        "Q of Diamonds","K of Diamonds"};
int cardIndex = (int)(Math.random()*13);          //this yields 0-12

System.out.println("You drew card index #"+cardIndex+" which is the "+
      deckOfCards[cardIndex]);

if(deckOfCards[cardIndex].equals("A of Diamonds"))
   System.out.println("Woohooo!!!!");

deckOfCards[cardIndex]=null;
   System.out.println(deckOfCards[cardIndex]);

cardIndex = (int)(Math.random()*13);          //this yields 0-12
if(deckOfCards[cardIndex]==null)
   System.out.println("You drew the same card again!!!");
else
   System.out.println("Your drew card index #"+cardIndex+
         " which is the "+deckOfCards[cardIndex]);
```

Sep 12-7:31 PM

Reminder:

If you ever use an index number that doesn't exist ...

ArrayIndexOutOfBoundsException

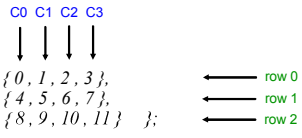
error is thrown!

Oct 29-11:04 AM

Returning to 2-D Arrays ...

Can be thought of as a table (rows/columns)

```
int myTable[ ][ ] = new int[3][4] ;          // row x column
```



Oct 29-7:27 AM

Returning to 2-D Arrays ...

Can be thought of as a table (rows/columns)

```
int myTable[ ][ ] = new int[3][4] ;          // row x column
```

How to fill the array ...

```
for(int i=0 ; i<myTable.length ; i++)
   for(int j=0 ; j<myTable[i].length ; j++)
      myArray[i][j]=i*j;
```

0	0	0	0
0	1	2	3
0	2	4	6

Oct 29-7:27 AM

How to "traverse" through arrays:

1D Arrays

Option #1: For-loop `for(i=0 ; i<theArray.length ; i++)`

Option #2: Extended `for(int element : theArray)`

Not for editing or replacing elements!

Oct 29-10:06 AM

How to "traverse" through arrays:

2D Arrays

Option #1: For-loop
```
for(i=0 ; i<theArray.length ; i++)
   for(j=0 ; j<theArray[i].length ; j++)
```

Option #2: Extended
```
for(int[] row : theArray)
   for(int element : row)
```

Not for editing or replacing elements!

Oct 29-10:06 AM

Things to be working on ...

* Wrap up Unit 3 WS02 - 2D Arrays
* Work on Unit 3 WS03 - More With Arrays

Sep 21-8:00 AM