## Simple `int` ARRAY

**An array is a data structure used to implement a list object, where the elements in the list are of the same type; for example, a class list of 25 test scores, a membership list of 100 names, or a store inventory of 500 items.**

**For an array of N elements in Java, index values go from 0 to N−1. Individual elements are accessed as follows: If `arr` is the name of the array, the elements are `arr[0]`, `arr[1]`, . . . , `arr[N-1]`.**

**In Java, an array is an object; therefore, the keyword `new` must be used in its creation. The size of the array remains fixed once it has been created. As with `String` objects, however, an array reference may be reassigned to a new array of a different size.**

**All of the following are equivalent:**

| | | |
|---|---|---|
| `double[] data = new double[25];` | `double data[] = new double[25];` | `double[] data;`<br>`data = new double[25];` |

**When arrays are declared, the elements are automatically initialized to zero for the primitive numeric data types (`int` and `double`), to `false` for `boolean` variables, or to `null` for object references.**

**The following are equivalent:**

| | |
|---|---|
| `int[] coins = new int[4];`<br>`coins[0] = 1;`<br>`coins[1] = 5;`<br>`coins[2] = 10;`<br>`coins[3] = 25;` | `int[] coins = {1, 5, 10, 25};`<br><br>***this is the one case where `new` is not required to create an array** |

**The length of an array is given by the final public instance variable `length`.**

**For example, the code shown below shows a loop that adds up the values of `coins` array:**

```
int total = 0;
for( int k = 0; k < coins.length; k++ )
    total += coins[k];
```

**Important:** **`length` is not a method and therefore is not followed by parenthesis. Contrast this with `String` objects, where `length` is a method and must be followed by parenthesis.**

## for-each loop

**Use a for-each loop whenever you need access to every element in an array without replacing or removing any elements. Use a for loop in all other cases: to access the index of any element, to replace or remove elements, or to access just some of the elements.**

**Important:** **Do not use a `for-each` loop to remove or replace elements of an array.**

**The example shown returns the number of even integers in array `arr` if integers.**

```
public int countEven( int[] arr )
{
   int count = 0;
   for( int num : arr )
      if( num % 2 == 0 ) count++;
   return count;
}
```

**Since arrays are treated as objects, passing an array as a parameter means passing its object reference. No copy is made of the array. Thus, the elements of the actual array can be accessed, and modified.**

**The example shown returns the index of the smallest element in array `arr`.**

```
public int findMin( int[] arr )
{
   int min = arr[0];
   int minIndex = 0;
   for( int k = 1; k < arr.length; k++ )
      if( arr[k] < min )
      {
         min = arr[k];
         minIndex = k;
      }
   return minIndex;
}
```

**Consider a simple Deck class in which a deck of cards is represented by the integers 0 to 51.**

```
public class Deck
{
    private int[] myCards;
    public final int NUMCARDS = 52;

    public Deck()
    {
        myCards = new int[NUMCARDS];
        for( int k = 0; k < NUMCARDS; k++ ) myCards[k] = k;
    }

    public void writeDeck()
    {
        int c = 0;
        int r = 0;
        for( int card : myCards )
        {
            if( c%13 == 0 && r != 0 )
              System.out.println();
            System.out.printf( "%-4d", card );
            c++;
            r++;
        }
    }

    private void swap( int[] arr, int i, int j )
    {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }

    public void shuffle()
    {
        int index;
        for( int i = NUMCARDS-1; i > 0; i-- )
        {
            index = (int)(Math.random() * (i + 1));
            swap( myCards, i, index );
        }
    }
}


//Driver
public class DeckMain
{
    public static void main( String args[] )
    {
        Deck d = new Deck();
        d.shuffle();
        d.writeDeck();
    }
}
```