The problem with resizing arrays and the motivation for ArrayList

There's no way to resize arrays

ArrayList allows for this

```
int[] array = new int[10];
// … fill the array …
int[] newArray = new int[20];
for(int i = 0; i < array.length; i++) {
    newArray[i] = array[i];
}
array = newArray;

array[10] = 5;
```

Constructing an ArrayList - Parameterization

ArrayLists must be lists of Objects

```
int[] array = new int[10];
ArrayList list = new ArrayList( );  // may produce warning about “parameterization”
                                    // storing generic Objects
ArrayList<Integer> intList = new ArrayList<Integer>( );

ArrayList<int> intList2 = new ArrayList<int>( ); // bad, doesn’t work ⇒ int is not an Object
```

Getting ArrayList length

```
int arrLength = array.length;
int arrayListLength = intList.size( );
```

Accessing ArrayList Values

```
int fourthNumber = array[3];
int fourthNumberInList = intList.get(3);
```

Adding / Putting Values into an ArrayList

```
array[3] = 6;
intList.set(3, 6); // index, value
                              // index you specify must be 0 ⇒ arraylist.size( )

int listSize = intList.size( ); // listSize == 4
intList.set(4, 1); ⇒ intList.add(1);

intList.add(6);  // adding the value to the end of the ArrayList
```

Generic Object ArrayList

```
ArrayList list = new ArrayList( );
list.add( (Object) new Integer(5) );
list.add( (Object) new Tokenizer(“arged” , “Tokenizers are useful”) );
list.add( (Object)  (“Hello”) );
```