7.1 - Arrays

An array is an ordered list of values.

The entire array has a single name. Each value has a numeric index.

An array of size N is indexed from zero to N-1.

Note that an array name is a pointer to the front element.

7.2 - Declaring Arrays

Some examples of array declarations:

```java
int[] scores = new int[100];
int[] units = {147, 323, 89, 269, 97, 114};
float[] prices = new float[500];
boolean[] flags;
flags = new boolean[20];
char[] codes = new char[1750];
char[] letterGrades = {'A', 'B', 'C', 'D', 'F'};
```

The iterator version of the for loop can be used to process array elements:

```java
for (int aScore : scores)
    System.out.println(aScore);
```

7.2 - Bounds Checking

How to avoid it?

```java
for (int index=0; index < scores.length; index++)
    scores[index] = scores[index-1] + scores[index+1];
```

7.2 - LetterCount.java

```java
//********************************************************************
// LetterCount.java demonstrates the relationship between arrays and strings.
//********************************************************************
import java.util.Scanner;
public class LetterCount {
    public static void main(String[] args) {
        final int NUMCHARS = 26;
        Scanner scan = new Scanner(System.in);
        int[] upper = new int[NUMCHARS];
        int[] lower = new int[NUMCHARS];
        char current; // the current character being processed
        int other = 0; // counter for non-alphabets
```

(more...)
7.2 - Arrays as Parameters

- An entire array can be passed as a parameter to a method
- Like any other object, the reference to the array is passed, making the formal and actual parameters aliases of each other
- Therefore, changing an array element within the method changes the original (called "by reference")
- This can also be a source of errors – be careful!

Copying and Comparing Arrays

- What happens here?
  myData = yourData;
- What is printed here?
  int[] input = {1, 2, 3, 5, 6};
  int[] output = {1, 2, 3, 5, 6};
  if (input == output)
    System.out.println("same");
  else
    System.out.println("different");
- How do we copy the contents of myData into yourData?
- How do we check if two arrays contain the same info?

Methods can have arrays as input

```java
public static int sumElements (int[] numArray) {
  int sum=0;
  for (int i=0; i<numArray.length; i++)
    sum = sum + numArray[i];
  return sum;
}
```

//... in main

```java
int[] myData = {1, 2, 3, 4, 5};
int result = sumElements(myData);
```
Methods can have arrays as output

```java
public static int[] createNumArray (int size) {
    int[] newArray = new int[size];
    for (int i=0; i< size; i++)
        newArray[i] = i;
    return newArray;
}
```

//... in main

```java
int[] arrayC = createNumArray(20);
```

Call “by reference” can have side effects

```java
public static void setArray (int[] numArray, int num)
{
    for (int i=0; i < numArray.length; i++)
        numArray[i] = num;
    // does not explicitly return anything
}
```

//... in main

```java
int[] arrayB = new int[10];
setArray(arrayB, 4);
```