Lecture 20:

Wrapping up Methods:
parameter variables and their scope

Begin Arrays

Mar 3 2015
Parameter Variables

A parameter variable is used to store information that is being passed from the location of the method call into the method that is called.

```java
public class Class1
{
    public static void method1()
    {
        double x;  // x contains // some info
        ....
        Class2.method2( x );
        // Pass (give) info
        // stored in x to
        // method2 !!!
        ....
    }
    ....
}

public class Class2
{
    public static void method2(double paramVar)
    {
        statements that operate on paramVar ....
        ....
    }
}
```

method1 is the caller

method2 is the callee
Lifetime (scope) of parameter variables

The scope of parameter variables is equivalent to declaring and initializing a local variable at the beginning of the code block inside the called method.

```java
public class ToolBox
{
    public static double min(double a, double b){
        // Body of method "min"
        double m = 0; //local var
        if ( a < b ){
            m = a;
        }
        else{
            m = b;
        }
        return(m);
    }
}
```

Calling min(10,32) is equivalent to the following:

```java
{double a = 10, b = 32;
 double m = 0; //local var

    if ( a < b ){
        m = a;
    }
    else{
        m = b;
    }
}
```
Introduction to **array**: why use arrays?

**Problem:**

Write a program that reads in and stores 5 numbers, and after reading in the numbers, computes the average of the numbers

**Potential Solutions**

Avg1.java

Avg2.java
Introducing Arrays

An **array** is a data structure that represents a collection of data with the *same data type*.

```
double[] myList = new double[10];
```

```
<table>
<thead>
<tr>
<th>Array element at index 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>myList[0]</td>
</tr>
<tr>
<td>myList[1]</td>
</tr>
<tr>
<td>myList[2]</td>
</tr>
<tr>
<td>myList[3]</td>
</tr>
<tr>
<td>myList[4]</td>
</tr>
<tr>
<td>myList[5]</td>
</tr>
<tr>
<td>myList[6]</td>
</tr>
<tr>
<td>myList[7]</td>
</tr>
<tr>
<td>myList[8]</td>
</tr>
<tr>
<td>myList[9]</td>
</tr>
</tbody>
</table>
```

Element value
Declaring and creating array variables

Syntax:

```plaintext
datatype[] arrayVar;
```

Example:

```plaintext
double[] a;
a = new double[5];
```

Example:

```plaintext
double[] a = new double[5];
```
Default values

When an array is created, its elements are assigned the default value of

- 0 for the numeric primitive data types,
- '\u0000' for char types, and
- false for boolean types.
Indexed variables

The array elements are accessed through the index. The array indices are 0-based, i.e., it starts from 0 to arrayRefVar.length - 1.

In the example a holds five double values and the indices are from 0 to 4. (this should be familiar from your work with strings)