Lesson 3…..Simple *String* Operations

In this lesson we will learn just a few of the things we can do with *String*s.

Concatenation:

First and foremost is **concatenation.** We use the plus sign, +, to do this. For example:

```
String mm = "Hello";
String nx = "good buddy";
String c = mm + nx;
System.out.println(c);  //prints Hellogood buddy…notice no space between o & g
```

The above code could also have been done in the following way:

```
String mm = "Hello";
String nx = "good buddy";
System.out.println(mm + "  " + nx);  //prints Hello good buddy…notice the space
```

We could also do it this way:

```
System.out.println("Hello" + "  good buddy");  // prints Hello good buddy
```

The *length* method:

Use the ***length( )*** method to find the number of characters in a *String*:

```
String theName = "Donald Duck";
int len = theName.length( );
System.out.println(len);  //prints 11…notice the space gets counted
```

Right now we don't see much value in this length thing…just wait!

A piece of a *String (substring*):

We can pick out a piece of a *String*…**substring**

```
String myPet = "Sparky the dog";
String smallPart = myPet.substring(4);
System.out.println(smallPart);  //prints ky the dog
```

Why do we get this result? The various characters in a *String* are numbered starting on the left with 0. These numbers are called **indices**. (Notice the spaces are numbered too.)

S p a r k y t h e d o g … so now we see that the 'k' has **index** 4 and we go from
0 1 2 3 4 5 6 7 8 9 10 11 12 13 k all the way to the end of the string to get "ky the dog".

A more useful form of *substring*:

But wait! There's another way to use ***substring***

```
String myPet = "Sparky the dog";
String smallPart = myPet.substring(4, 12);
System.out.println(smallPart);  //prints ky the d
```

How do we get ***ky the d***? Start at *k*, the 4th index, as before. Go out to the 12th index, 'o' in this case and pull back one notch. That means the last letter is *d*.

Conversion between lower and upper case:

toLowerCase converts all characters to lower case (small letters)

```
String bismark = "Dude, where's MY car?";
System.out.println( bismark.toLowerCase( ) );  // prints dude, where's my car?
```

toUpperCase converts all characters to upper case (capital letters)

```
System.out.println( "Dude, where's My car?".toUpperCase( ) );
                                          //prints DUDE, WHERE'S MY CAR?
```

Note: *length*, *substring*, *toLowerCase*, and *toUpperCase* are all **methods** of the *String* class. There are other methods we will learn later.

Concatenating a *String* and a numeric:

It is possible to concatenate a *String* with a numeric variable as follows:

```
int x = 27;
String s = "Was haben wir gemacht?";  //German for "What have we done?"
String combo = s + " " + x;
System.out.println(combo);  //prints Was haben wir gemacht? 27
```

Escape sequences:

How do we force a **quote** character (") to printout…. or, to be part of a *String*. Use the **escape sequence**, \", to print the following (note escape sequences always start with the \ character…see Appendix B for more on escape sequences):

```
What "is" the right way?
```

```
String s = "What \"is\" the right way?";
System.out.println(s);  //prints What "is" the right way?
```

Another **escape sequence,** \n, will create a **new line** (also called **line break**) as shown below:

```
String s = "Here is one line\nand here is another.";
System.out.println(s);
```

Prints the following:

```
Here is one line
and here is another.
```

The **escape sequence**, \\, will allow us to print a backslash within our *String*. Otherwise, if we try to insert just a single \ it will be interpreted as the beginning of an escape sequence.

```
System.out.println("Path = c:\\nerd_file.doc");
```

Prints the following:

```
Path = c:\nerd_file.doc
```

The **escape sequence**, \t, will allow us to "tab" over. The following code tabs twice.

System.out.println("Name:\t\tAddress:");

Prints the following:

```
Name:           Address:
```

Exercise on Lesson 3

1. Write code in which a *String* variable *s* contains "The number of rabbits is". An integer variable *argh* has a value of 129. Concatenate these variables into a *String* called *report*. Then print *report*. The printout should yield:
   ```
   The number of rabbits is 129.
   ```
   Note that we want a period to print after the 9.

2. What is the output of *System.out.println(p.toUpperCase( ));* if p = "Groovy Dude"?

3. Write code that will assign the value of "Computer Science is for nerds" to the *String* variable *g*. Then have it print this *String* with nothing but "small" letters.

4. What will be the value of *c*?
   String c;
   String m = "The Gettysburg Address";
   c = m.substring(4);

5. What will be the value *c*?
   String b = "Four score and seven years ago,";
   String c = b.substring(7, 12);

6. What is the value of *count*?
   int count;
   String s = "Surface tension";
   count = s.length( );

7. Write code that will look at the number of characters in *String m = "Look here!";* and then print
   ```
   "Look here!" has 10 characters.
   ```
   Use the *length( )* method to print the 10 ….you must also force the two quotes to print.

8. How would you print the following?
   ```
   All "good" men should come to the aid of their country.
   ```

9. Write code that will produce the following printout using only a single *println( )*.

```
Hello
Hello again
```

10. Write code that will produce the following printout.

```
A backslash looks like this \, …right?
```

11. What is output by the following?

```
String pq = "Eddie Haskel";
int  hm = pq.length( );
String ed = pq.substring(hm - 4);
System.out.println(ed);
```

12. Which character is at the 5th index in the String "Herman Munster"?

Project… Name that Celebrity

Create a new project called *NameThatCelebrity* in which only partially recognizable names of celebrities are to be produced. In a real implementation of this game, the idea is for a contestant to be able to guess the real name of the celebrity after the first two and last three letters are dropped from the name. We have been given the task of testing the feasibility of this idea by producing the following printout:

```
Allan Alda>>>lan A
John Wayne>>>hn Wa
Gregory Peck>>>egory P
```

Begin your code within the *main* method as follows:

```
String s1 = "Allan Alda";
String s2 = "John Wayne";
String s3 = "Gregory Peck";
```

Apply the *length* and *substring* methods to these *String*s to produce the above printout.