Concatenation

Concatenation is when two or more Strings are put together. Some people call this "adding Strings together".

When two Strings are concatenated together a third String is created that is the combination of both Strings, that is, it is the first String followed by the second String. For example:

```java
public class TestConcatenstion {
    public static void main(String[] args) {
        System.out.println("John" + "Lennon");
    }
}
```

The output displayed in the console window is

    JohnLennon

Notice that quotes are not displayed (because they denote the beginning and end of the String, they are not part of the String.) There is no space between John and Lennon. The two Strings are concatenated together (jammed together) just as they are denoted inside the quotes. If you wanted a space between John and Lennon you would have to place a space inside the quotes. For example:

```java
System.out.println("John " + "Lennon");
```

The output would now be

    John Lennon

because there was a space placed after John inside the quotes.

You can concatenate together as many Strings as you want. For example:

```java
System.out.println("John " + "Paul " + "George " + "and Ringo");
```

which produces the output

    John Paul George and Ringo

After looking at this, this doesn't seem to be anything special. Why not just put them all in quotes like this?

```java
System.out.println("John Paul George and Ringo");
```

I agree!!! Here is an example showing the real reason we use concatenation:

```java
System.out.println("Bob's I.Q. is " + 115);
```

The output is

    Bob's I.Q. is 115
There is a concatenation but it is between the String "Bob's I.Q. is " and the int 115. What's going on?

Java can only add ints, and clearly "Bob's I.Q. is " is not an int, therefore the plus sign must mean concatenation, but 115 is an int not a String! What is Java to do?

Java can convert any primitive data type or object into a String. So in the example above the int 115 is converted into the String "115", then the concatenation is performed.

Here is an complicated expression and it's output. Can you see how Java interprets this expression?

```java
System.out.println(3 + " + " + 7 + " = " + (3 + 7));
```

Output is:

```
3 + 7 = 10
```

Java first converts the int 3 into the String "3" and concatenates the "3" and " + " together to produce the String "3 + ".

Next Java converts the int 7 into the String "7" then concatenates "7" with "3 + " to produce the String "3 + 7".

The above computed String is concatenate with the String " = " to produce "3 + 7 = ".

Since there are parenthesis around (3 + 7), the two ints 3 and 7 are arithmetically added together before any concatenation is done. The sum is converted to a String "10", which is then concatenated with previous String computed yielding the String "3 + 7 = 10".