UNIT 6 L 1 Strings and String manipulation

Processing, being a programming language based on Java, follows the same rules of object oriented programming. The variables studied so far are in fact categorized in 2 groups -primitive and Class. Strings are a class and as such subject to different rules regarding their use. Text is stored differently in the computer memory and any changes or queries we want to make are accomplished with special String methods.

The string methods are very specific in what they do and each has its own syntax, which needs to be followed strictly.

I. **EQUALS()**- Strings are compared using **EQUALS** method. The method compares 2 words letter by letter, including capitalization, and determines if two words (or sentences) are identical. Attempting to compare two strings using **==** leads to a false result as the only thing that happens when using == with strings is the attempt to compare if two strings reside in the same location in memory.

1. Syntax

   word1.equals(word2)

2. Return value = true/false. False result may be received despite two words being identical as they may have different capitalization and/or spacing before or after the word or even in the middle.

Example 1

String w = "Mackenzie";	(a) w.equals("Mackenzie") will return **true** (b) w.equals("mackenzie") will return **false** because 'M' ≠ 'm' (c) w.equals("Mackenzie " will return **false** because of the blank in the string "Mackenzie "

II. compareTo() Refers to comparing two words to determine their lexicographic order (i.e. the order they would be written in a dictionary. Thus we can determine if one word precedes the other.**The two strings are compared to each other left to right, character by character**.

The method returns one or three values :

-1 (negative) first string preceeds the second 0 both strings are identical 1 (positive) the second string precedes first	a) "cab".compareTo("car") < 0 because 'b' < 'r' (b) "Car".compareTo("car") < 0 because 'C' < 'c' (c) "27".compareTo("186") > 0 because '2' > '1' (d) "car".compareTo ("cart") < 0 because "car" is only the first part of "cart"

Test in Processing by printing each comparison

The compareTo method can be used to implement any of the six relational operators (<, <=, >, >=, ==, or !=) by writing expressions of the form $<string_1>.compareTo(<string_2>) <op> 0$

III.substring() – creates a smaller string from the original string. Defined by the values incud

Word.subsrting(startPos,stopPos);

IV.Concat() – both concat() and (+) join two strings

Practice

1. Type the following statements in new Processing file. Indicate the output.
   String firstName = “Mickey";
   String lastName = “Mouse”;

println (firstName + lastName);	MickeyMouse
println (firstName.substring(2));	ckey
println (firstName.substring(4));	ey
println (lastName.substring (2,4));	us
println (lastName.length( ));	5
println (firstName.charAt (0));	M
println (firstName.charAt (firstName.length( )-1));	y
println(firstName.concat(lastName));	MickeyMouse
println(lastName.toUpperCase());	MOUSE
println(lastName.toLowerCase());	mouse