CS163/164 – Strings Worksheet

1. Review the following String methods.

indexOf()	***indexOf***(item) gets index of first item occurrence in a string, else -1. Item may be char, String variable, or string literal.
	
	indexOf(item, indx) starts at index indx.
	
	lastIndexOf(item) finds the last occurrence of the item in a string, else -1.
		`// userText is "Help me!"`
		`userText.indexOf('p')  // Returns 3`
		`userText.indexOf('e')  // Returns 1 (first occurrence)`
		`userText.indexOf('z')  // Returns -1`
		`userText.indexOf("me")  // Returns 5`
		`userText.indexOf('e', 2)  // Returns 6 (starts at index 2)`
		`userText.lastIndexOf('e') // Returns 6 (last occurrence)`

substring()	***substring***(startIndex) returns substring starting at startIndex.
	
	substring(startIndex, endIndex) returns substring starting at startIndex and ending at endIndex - 1. The length of the substring is given by endIndex - startIndex.
		`// userText is "http://google.com"`
		`userText.substring(7)  // Returns "google.com"`
		`userText.substring(13)  // Returns ".com"`
		`userText.substring(0, 7)  // Returns "http://"`
		`userText.substring(13, 17) // Returns ".com"`
		
		`// Returns last 4: ".com"`
		`userText.substring(userText.length() - 4, userText.length())`

concat	***concat***(moreString) creates a new String that appends the String moreString at the end.	`// userText is "Hi"`
		`userText = userText.concat(" friend"); // Now "Hi friend"`
		`newText = userText.concat(" there");`
		`// newText is "Hi friend there"`

replace()	***replace***(findStr, replaceStr) returns a new String in which all occurrences of findStr have been replaced with replaceStr.
	
	replace(findChar, replaceChar) returns a new String in which all occurrences of findChar have been replaced with replaceChar.
		`// userText is "Hello"`
		`userText = userText.replace('H', 'j'); // Now "jello"`
		`// userText is "You have many gifts"`
		`userText = userText.replace("many", "a plethora of");`
		`// Now "You have a plethora of gifts"`
		`// userText is "Goodbye"`
		`newText = userText.replace("bye"," evening");`
		`// newText is "Good evening"`

str1 + str2	Returns a new String that is a copy of str1 with str2 appended.
	
	str1 may be a String variable or string literal. Likewise for str2. One of str1 or str2 (not both) may be a character.
		`// userText is "A B"`
		`myString = userText + " C D";`
		`// myString is "A B C D"`
		`myString = myString + '!';`
		`// myString now "A B C D!"`
		`myString = myString + userText;`
		`// myString now "A B C D!A B"`

str1 += str2	Shorthand for str1 = str1 + str2.	`// userText is "My name is "`

2. Considering your understandings from your review. What are the exact outputs in the program below?

```
public class StringMethods {
    public static void main(String args[]){
        String fullDayString = "09/30/2022";
        String month = fullDayString.substring(0,fullDayString.indexOf("/"));
        String day = fullDayString.substring(3,5);
        String year = fullDayString.substring(6);
        System.out.println("Day: " + day);
        System.out.println("Month: " + month);
        System.out.println("Year: " + year);
    }
}
```

For the activities below, write your solution in English first, think about each step that needs to be done and after that you can transform your solution to Java code.

Complete the class StringMethods provided to:

3. Write a method that receives a String and build a new String containing only the characters that are letters.

4. Write a method that receives a String, build a new String containing all characters that are digits, transforms and return that String into a long.

5. Write a method that receives a String and a char and returns the number of times that the char appears in the String.