Computer Science 12 AP

String Theory

int length()	*String* substring(int from)	*int* indexOf(String s)
boolean equals(String other)	*String* substring(int from, int to)	int indexOf(String s, int pos)
int compareTo(String other)	the + operator	

Create a class called 'StringTheory'. Code the following methods and then test them out using a runner class.

1. *public boolean sameWords(String a, String b, String c)* that returns true if the three words all equal each other. False otherwise.

2. *public String alphaOrder(String a, String b)* that returns the word (of the two) that would be alphabetically first.
   Hint: use compareTo. Just make it work within this method!

3. *public void chunks(String s)* that will print out all the 3 letter groups as the example shows:
   chunks("abcdefgh") would output abc, bcd, cde, def, efg, fgh. Try to accomplish this using a loop!
   Hint: You will probably use the line s.substring(i, i+3) in your code. Make sure your loop doesn't go too far past the end of the string!

4. *public int ENum(String s)* that will return the number of "e" 's in the word or sentence.
   Hint: use a loop to cycle through each letter of the string. Check if it's an "e". If so, add to a counter.

5. *public int howMany(String s)* that will return the number of times the word *"the"* appears in the String s. You should test your functions with a sentence that contains various occurrences of the word *the*.
   Hint: use a loop to cycle through each three letter combo in the sentence you enter.

6. *public void DateSplit(String date)* that will split the String *date* into three parts based upon the character "-". The String *date* will always be in the format 2007-10-28, or 08-5-17. (year-month-day). Print the year, month, and day on a separate line.

7. *public String pad5(int x)* that will pad a number with zeros and return a 5 digit string. For example pad5(15) would return "00015", pad5(182) would return 00182 , pad5(12345) would return "12345". Assume that x is 5 or less digits.

8. *public void individualWords(String sent)* that will print each word in sentence on a new line. Assume that a space bar " " separates every word and the sentence ends with a "."
   Hint: Talk with partners to come up with a routine you agree will work. Look for space bars. Keep track of where you last found a space bar. Keep track of where the next one is. Keep looping until you hit the "." Character. Good luck.

9. *public String makeName()* that will construct and return a name that is 4 letters long.

   Hint: one idea is to create a string that contains all the consonants in the alphabet. Then create a string that contains all the vowels in the alphabet. Randomly pick a character from each list and join them with the + operator so that you form names like 'babi', 'kujo', 'tamo', … Yes, the names will all sound cool! The first successful name you print out will be your Computer Science AP nickname.

10. *public String pad5(int[] nums)* that will return a string representing ALL the numbers in the array padded with zeros to form a long string of 5 digit numbers.
    For example, an array passed in set to
    {51, 12, 182, 8, 631} would become "000510001200182000800631"
    Hint: use your method pad5(int x) that you already wrote earlier!

11. *public boolean containsNumeric(String s)* that returns true if the string passed in contains at least one numerical character. For example
    containsNumeric("ab3sj") would return true
    containsNumeric("abcdef") would return false