Code Snippit Quiz 1

1. A loop that will assign 10 random integers to an array *nums* ranging from 5 to 99 (inclusive).
2. Display the *num* values that are evenly divisible by 3.
3. Sort *num* in ascending order.
4. Given String *myString* = “John Champe AP Computer Science focuses on Java”, display the final word (only) using the *substring* method.
5. Display the number of words in *myString*
6. Display the number of times the letter “o” occurs within the *myString*

Code Snippit Quiz 2 – Arrays

1. Create a static method that receives a 2 dimensional integer array (int [][]) and returns the sum of all values.
2. Create a static method that receives a 2 dimensional integer array (int [][]) and returns the minimum value within the array.
3. Create a static method that receives a 2 dimensional integer array (int [][]) and returns an ArrayStats object that provides the following statistics about the input array: number of rows (getRowCount()), number of columns (getColCount())
4. Create static methods that receive a 2 dimensional integer array (int [][]) and return an List for the column or row designated. Return null if the requested row/column is out of bounds => static List columnToList(int[][] inAry, int col), static List getRowList(int[][] inAry, int row)
5. Create a static method that receives an integer array (int[]) and returns a sorted ArrayList (ascending sort) => static List sort(int[] inAry)

Code Snippit Quiz 3 – String Logic[[1]](#footnote-1)

Consider a guessing game in which a player tries to guess a hidden word. The hidden word contains only capital letters and has a length known to the player. A guess contains only capital letters and has the same length as the hidden word.

After a guess is made, the player is given a hint that is based on a comparison between the hidden word and the guess. Each position in the hint contains a character that corresponds to the letter in the same position in the guess. The following rules determine the characters that appear in the hint.

|  |  |
| --- | --- |
| **If the letter in the guess is …** | **the corresponding letter in the hint is** |
| Same position within the word (match) | Show the matching letter |
| in the hidden word, but a different position | “+” |
| not in the hidden word | “\*” |

The HiddenWord class will be used to represent the hidden word in the game. The hidden word is passed to the constructor. The class contains a method, getHint, that takes a guess and produces a hint.

For example, suppose the variable puzzle is declared as follows.

HiddenWord puzzle = new HiddenWord("HARPS");

The following table shows several guesses and the hints that would be produced:

|  |  |
| --- | --- |
| **Call to getHint** | **String returned** |
| puzzle.getHint("AAAAA") | "+A+++" |
| puzzle.getHint("HELLO") | "H\*\*\*\*" |
| puzzle.getHint("HEART") | "H\*++\*" |
| puzzle.getHint("HARMS") | "HAR\*S" |
| puzzle.getHint("HARPS") | "HARPS" |

Write the complete HiddenWord class, including any necessary instance variables, its constructor, and the method, getHint, described above. You may assume that the length of the guess is the same as the length of the hidden word.

1. https://secure-media.collegeboard.org/digitalServices/pdf/ap/ap15\_frq\_computer\_science\_a.pdf [↑](#footnote-ref-1)