Array Coding Assignment – `truncateString`

`truncateString`

Data validation is the process of checking if data is accurate, complete, and meets specific rules or requirements before it's used. The data may be checked for numerical range, length, format, etc. For example, a numerical month must be in the range of 1 through 12. Or a valid email is required to contain, among other requirements, an "at" symbol (@). The storage of a string, such as in a database, often requires a restriction put on its length.

To truncate a string, we can use the `String` instance method `substring`. To restrict the length of a string `s` to a length given by a constant named `MAX_LENGTH`, we would write: "`s.substring(0, MAX_LENGTH);`".

However, this line of code will cause an error if the string is shorter than `MAX_LENGTH`, so there needs to be an `if` statement to separate the case where `s` is less than `MAX_LENGTH`.

1. Define a class field `MAX_LENGTH` to a value of 5, then write a method according to the following specifications:
   - named `truncateString`
   - takes in a parameter named `label` of type `String`
   - returns a string that is equal to `label` if the length of the string is equal or shorter than than `MAX_LENGTH` characters
   - returns a string that is the contents of `label` truncated to a length of `MAX_LENGTH` characters if the length of `label` exceeds a length in characters of `MAX_LENGTH`.

Write the completed code in the box below.

```
// class field
public static final int MAX_LENGTH = 5;
// method truncateString
public static String truncateString(String label) {
   if (label.length() <= MAX_LENGTH) {
      // already fewer or equal to MAX_LENGTH
      return label;
   } else {
      // truncate to MAX_LENGTH characters
      return label.substring(0, MAX_LENGTH);
   }
}
```

`fibonacci`

The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones. Mathematically, this can be expressed as: $F(n) = F(n-1) + F(n-2)$. However, we need to define how the sequence starts, which may be either with “`0`, `1`, …”, or, for the “Lucas Sequence” variance, with “`1`, `1`, …”.

2. Write the Java code (writing directly in the `main` method is fine) that creates an array containing the Fibonacci numbers by following the instructions given below.
   - Define a class field or local constant named `ARRAY_LENGTH` to have a value of `10`
   - Declare and instantiate an array of length equal to `ARRAY_LENGTH` to store the Fibonacci sequence
   - Manually initialize the first two elements of the array to both equal 1.
   - Use a `for` loop to calculate and write the Fibonacci values into the remainder of the array.
   - Use an <u>enhanced</u> `for` loop to print each element of the array separated by a space character.

Write the completed code in the box below.

```java
int[] fibonacci = new int[ARRAY_LENGTH];
// Initialize the first two Fibonacci numbers
fibonacci[0] = 1;
fibonacci[1] = 1;
// Fill the array with Fibonacci numbers
for (int i=2; i<fibonacci.length; i++) {
    fibonacci[i] = fibonacci[i-1] + fibonacci[i-2];
}
// Print the Fibonacci numbers
System.out.print("Fibonacci numbers: ");
for (int num : fibonacci) {
    System.out.print(num + " ");
}
System.out.println();
```