Homework 2: Compute Check Digits

Due date: September 7, 2020 (Mon)

For this homework, you will make a class called CheckDigit that computes the check digit for a 12-digit UPC number. The program is given command line input consisting of the first 11 digits of a 12-digit UPC number. For more background on UPC numbers check out the Wikipedia article on the Universal Product Code (especially the section on check digits) or other resources on the web.

Specification

- Accept an argument from the command line that is the first eleven digits of a 12-digit UPC from any typical product sold in the United States (without spaces or hyphens or anything else but the eleven digits).
- Compute the UPC's 12th digit (the check digit)
- Print out two lines of output
    - The first eleven digits given to you on the command line
    - The 12th digit that you computed
- Compile and test your program
- Follow the Java Coding Conventions as described in http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html.
- There must be a JavaDoc comment at the beginning of your Java file indicating that you are the author of the file (so this comment must include an @author tag).

Implementation Hints

- For any 12-digit UPC of the format: $a\ bcdef\ ghijk\ x$ (where a-k and x are the digits of the UPC and spaces are added only to enhance readability), the check digit x can be computed from the digits a-k using the following formula:

$$x = (10 - (3a + b + 3c + d + 3e + f + 3g + h + 3i + j + 3k) \bmod 10) \bmod 10$$

- You do not need to validate that the command line input given to CheckDigit is a number. (Specifically, it's acceptable if your program blows up when given an invalid input. You may assume that 11 digits will be provided in the command line argument whenever we test your program. Also, you don't need to catch the `NumberFormatException` that would be generated by `Integer.parseInt()` or `Long.parseLong()` if non-numeric characters were to be provided.)

Turning-in Your Work

Submit your `CheckDigit.java` file using AutoLab (https://autolab.andrew.cmu.edu).

Grading

AutoLab will grade your assignment as follows:

- Test cases on autolab (17 tests @ 5 points each)
- Follows coding conventions (up to 10 points)
    - We'll deduct one point for each coding convention issue detected.

- Author JavaDoc comment at beginning of file (5 points)

The Checkstyle (http://checkstyle.sourceforge.net) is being used in Autolab to check whether you follow the coding conventions or not. You are more than welcome to download this in your local machine to check your code on your own before submitting it onto Autolab. We use the **sun_checks.xml** configuration file.

AutoLab will show you the results of its grading within approximately one or two minutes of your submission. You may submit multiple times so as to correct any problems with your assignment. Autolab uses the last submission as your grade.