

# CS 1101 Comprehensive Lab 2

Released on: **Monday, October 22nd**

Due on: **Thursday November 8, 11:59pm**

How: submit zip file to Google classroom

What: A zip file named after you (LastNameFirstName-complab2.zip) containing a Java file as a solution to the below problems and a word document for the below pseudocode problems.

## Lab Description

This lab will be on methods, arrays (1D and 2D), and on repetition (using loops). In this lab, you will have to complete two activities. Here is what you have to do:

### Activity 1. [60 points]

**Method Luhn.** You will have to implement the Luhn Checksum Validation algorithm to check whether a credit card number is valid or fake. This is a widely used algorithm that checks that you enter a valid credit card number when you are trying to make a purchase.

Short description of the algorithm [*Think Like a Programmer: an Introduction to Creative Problem Solving* by V. A. Spraul]: Using the original number, double the value of every other digit, starting with the leftmost one, as shown below. Then add the values of the individual digits together (if a doubled value now has two digits, add the digits together – see below). The identification number is valid if the sum is divisible by 10 (i.e., the sum has to be a multiple of 10).

Example of execution of the algorithm:

Suppose you want to check that your credit card number is valid. Your credit card number is 8273 1232 7351 0569. Let's see how to check it:

8	2	7	3	1	2	3	2	7	3	5	1	0	5	6	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

First, you are going to double every other number, starting with the first number (here, it is number 8):

8x2	2	7x2	3	1x2	2	3x2	2	7x2	3	5x2	1	0x2	5	6x2	9
-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---

And you obtain:

16	2	14	3	2	2	6	2	14	3	10	1	0	5	12	9
----	---	----	---	---	---	---	---	----	---	----	---	---	---	----	---

But we do not want double digits, so for every number that now has double digits, we add these digits:

1+6	2	1+4	3	2	2	6	2	1+4	3	1+0	1	0	5	1+2	9
-----	---	-----	---	---	---	---	---	-----	---	-----	---	---	---	-----	---

And we obtain:

7	2	5	3	2	2	6	2	5	3	1	1	0	5	3	9
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Now we add all of these numbers:

$7 + 2 + 5 + 3 + 2 + 2 + 6 + 2 + 5 + 3 + 1 + 1 + 0 + 5 + 3 + 9 = 56$

56 is not a multiple of 10, so the credit card number was a fake...

To complete this activity, you will have to:

- 1/ Write the pseudocode of Method Luhn in the word document that you will submit along with the java file of this lab;
- 2/ Implement the method luhnCheck in the file CompLab2.java and add relevant code in the main method to test it; and
- 3/ Provide two examples of input and the corresponding output for Method Luhn – one valid and one invalid credit card number.

## Activity 2. [40 points]

**Method checkCustomers.** In this activity, imagine that you have a business that charges clients for some service. You record your clients' billing information as their names and credit card information, and you store this information in a 2-dimensional array (one row per client, 2 pieces of information per client: name and credit card number). You then proceed to charging your clients but it may happen that some credit card numbers are invalid, because they were wrongly inputted for instance. You want to make sure to keep track of the credit card numbers that are invalid, so you proceed by checking each credit card number for validity and you will write all erroneous information (credit card number along with name of client or name of business) to the terminal after the user has entered all clients' billing information.

For this method, you have to:

- 1/ Write the pseudocode of it in the word document that you will submit along with the java file of this lab; and
- 2/ Implement it in the file CompLab2.java and add relevant code in the main method to test it.

## Bonus Activity. [30 points]

**Method generateCCard.** Finally, in this bonus activity, imagine that you are the person in charge of issuing credit cards for new customers: you will have to generate card numbers for them. For this, you have to design and implement a method called generateCCard, which generates valid credit card numbers. This method does not take any input and returns a 16-digit valid credit card number. This number should change at each execution of the method (i.e., the generation should be random, yet valid).

If you choose to complete this bonus activity, you will have to:

- 1/ Write the pseudocode of it in the word document that you will submit along with the java file of this lab; and
- 2/ Implement it in the file CompLab2.java and add relevant code in the main method to test it.

## What do you have to turn in?

- 1/ The updated CompLab2.java file with your name as the first line of the code
- 2/ A full description of the methods of Activities 1 & 2 in a word document: as specified in each activity.