# CIS016-1 - Principles of Programming / CIS096-1 – Principles of Programming and Data Structures / PAT001-1 – Principles of Programming - 2019/2020
# Exercise Sheet 6

This exercise sheet is related to the Week 5 and previous lectures, so please consult the lecture notes when attempting the exercises. Please also download the example code. All can be found on BREO under "Guided Learning → Week 5".

## Exercise 1 Account Holders

1. Alter the **Account** class in the Week 5 example code to include two new instance members (variables); one for account holder's name, and the other for account holder's address. Each of these members should be supported with a set<name of variable> method and a get<name of variable> method. A new constructor method should also be created to initialise all variables including the two new ones.
2. Produce an orchestrating class which will ask the user for a name and an address (using the **Scanner** class), and then pass this data into the new object constructor method. Create an object from this input data and output all the object instance variables.
3. Produce an orchestrating class that creates three **Account** objects and displays all the data.

## Exercise 2 Gauss[1] would have loved it!

Write a program to input a whole number *n* using the **Scanner** class. If the number is less than 0, your program should print an error message. If the number is 0 or 1, the actual number should be printed. If the number is greater than 1, compute the sum of all integers between 1 and the given number *n* (1+2+...+*n*), using a for loop, and print this sum[2].

## Exercise 3 Ten Green Bottles in a Loop

Write a class **GreenBottles**. You find the lyrics of the traditional Ten Green Bottles song mentioned below on the next page.

1. Define a method **tenGreenBottles** that can print out the words to the song Ten Green Bottles. Use a loop for its implementation. (Hint: the final 2 verses may be tricky.)
2. In the same file define a function called **xGreenBottles** which when given the input x (as an integer parameter) will produce and return the words of the song for a given value of x green bottles (starting with x green bottles and removing 1 green bottle each

[1] https://en.wikipedia.org/wiki/Carl_Friedrich_Gauss

[2] Some of you may recognise that this sum can directly be calculated without a loop. But here we explicitly ask for a solution using a for loop!

time). Use a loop for its implementation.

3. Finally define a method called **xyGreenBottles** which will produce the words of a song where, starting with x green bottles, y bottles are removed every time until there are none left. Your method therefore takes two integer parameters, x and y. Use a loop for its implementation.

## Ten Green Bottles – Anon (Traditional)

10 green bottles standing on the wall,
10 green bottles standing on the wall,
And if 1 green bottle should accidently fall,
There'll be 9 green bottles standing on the wall.

9 green bottles standing on the wall,
9 green bottles standing on the wall,
And if 1 green bottle should accidently fall,
There'll be 8 green bottles standing on the wall.

8 green bottles standing on the wall,
8 green bottles standing on the wall,
And if 1 green bottle should accidently fall,
There'll be 7 green bottles standing on the wall.

7 green bottles standing on the wall,
7 green bottles standing on the wall,
And if 1 green bottle should accidently fall,
There'll be 6 green bottles standing on the wall.

6 green bottles standing on the wall,
6 green bottles standing on the wall,
And if 1 green bottle should accidently fall,
There'll be 5 green bottles standing on the wall.

5 green bottles standing on the wall,
5 green bottles standing on the wall,

# CIS016-1 - Principles of Programming / CIS096-1 – Principles of Programming and Data Structures / PAT001-1 – Principles of Programming - 2019/2020
# Exercise Sheet 6

And if 1 green bottle should accidently fall,
There'll be 4 green bottles standing on the wall.

4 green bottles standing on the wall,
4 green bottles standing on the wall,
And if 1 green bottle should accidently fall,
There'll be 3 green bottles standing on the wall.

3 green bottles standing on the wall,
3 green bottles standing on the wall,
And if 1 green bottle should accidently fall,
There'll be 2 green bottles standing on the wall.

2 green bottles standing on the wall,
2 green bottles standing on the wall,
And if 1 green bottle should accidently fall,
There'll be 1 green bottle standing on the wall.

1 green bottle standing on the wall,
1 green bottle standing on the wall,
And if 1 green bottle should accidently fall,
There'll be no green bottles standing on the wall.