Welcome to Lab 3!

This week, you are going to continue getting acquainted with the Java programming language. We have prepared a skeleton code for you to tinker with. This lab will give you the opportunity to increase your familiarity with:

- Input / output
- Variables, computations
- Algorithms
- Conditional statements

Have fun!

Ok, let's get started! Here are the two activities you will be working on.

You should expect to work about 2 to 3 extra hours outside the lab session to complete this assignment:

This includes completing the activities and taking the time to make sure that your submission is picture perfect!

Activity 1.

In this activity, you will get to practice on input, output, variables, and types. You will have to implement the solution code of the exercise that is described below:

Software Sales
A software company sells a package that retails for $99. Quantity discounts are given according to the following table:

Quantity:	**Discount:**
10 – 19	20%
20 – 49	30%
50 – 99	40%
100 or more	50%

Design a program that asks the user to enter the number of packages purchased. The program should then display the amount of the discount (if any) and the total amount of the purchase after the discount.

Example: Imagine a client wants to buy 25 software packages. The client will get a 30% discount. The regular price would have been: $2,475 but the discounted price is now: $1,732.5.

Pseudocode of the solution:

```
1. Ask the client for the number of packages s/he wants to buy
2. Retrieve this number using some kind of a listener and store it in an integer
   variable called numSoftware
3. If the value stored in numSoftware is 9 or less, then store the real value 0 in a
   variable called discount
4. If the value stored in numSoftware is between 10 and 19, then store the real
   value 0.2 in a variable called discount
5. If the value stored in numSoftware is between 20 and 49, then store the real
   value 0.3 in a variable called discount
6. If the value stored in numSoftware is between 50 and 99, then store the real
   value 0.4 in a variable called discount
7. If the value stored in numSoftware is 100 or more, then store the real value 0.5
   in a variable called discount
8. Compute the following: 99 * numSoftware * (1 – discount) and store this value
   in a real variable called discountedPrice
9. Display the value of the discount by outputting what is stored in your variable
   called discount
10.Display the price the client has to pay by outputting what is stored in your
   variable called discountedPrice
```

Another version would allow you to revise the price of each software package much more easily. It would read as follows (the modification is in bold):

```
1.  Ask the client for the number of packages s/he wants to buy
2.  Retrieve this number using some kind of a listener and store it in an integer
    variable called numSoftware
3.  If the value stored in numSoftware is 9 or less, then store the real value 0 in a
    variable called discount
4.  If the value stored in numSoftware is between 10 and 19, then store the real
    value 0.2 in a variable called discount
5.  If the value stored in numSoftware is between 20 and 49, then store the real
    value 0.3 in a variable called discount
6.  If the value stored in numSoftware is between 50 and 99, then store the real
    value 0.4 in a variable called discount
7.  If the value stored in numSoftware is 100 or more, then store the real value 0.5
    in a variable called discount
8.  Compute the following: 99 * numSoftware * (1 – discount) and store this value
    in a real variable called discountedPrice
9.  Display the value of the discount by outputting what is stored in your variable
    called discount
10. Display the price the client has to pay by outputting what is stored in your
    variable called discountedPrice
```

Instructions:

1. Design (5) scenarios of usage of your (pseudo)code: for this, provide a user input (number of software packages to be purchased) and show the expected output.
   a. Provide these scenarios in a docx file named “yourLastName-yourFirstName-lab3.docx”

   Note: *sets of scenarios that cover the most aspects of the code will be rewarded with extra points.*
2. Modify the provided code where prompted to make sure that the code follows either of the two above versions (specify as a comment in your code, which one you chose).

What you have to turn in:

1. The java file with the completed code for method called **SoftwareSales**.
2. A docx file in which you have described scenarios as prompted above.

Activity 2.

In this activity, you will get to practice some more on variables and type. You will have to implement the solution code of the exercise that is described below:

Trading Piggy Banks
Zoe lives in the US and has a piggy bank filled with US dollars.
Gonzague lives in France and has a piggy bank filled with Euros.
They want to trade the content of their piggy banks but they need the transferred amount to be also converted to their respected currencies.

You have to implement a piece of software that allows them to do just this.
More specifically, you should design a program that:

- asks Zoe the amount of dollars that she has
- asks Gonzague the amount of euros that he has
- displays these original amounts
- converts the amount of dollars of Zoe into euros and sets this amount to be the new amount that Gonzague owns
- converts the original amount of euros of Gonzague into dollars and sets this amount to be the new amount that Zoe owns

displays the new amounts along with their owners and respective currencies

Example:
If originally Zoe has $100 and Gonzague 200 euros, then at the end, Zoe should have the equivalent of 200 euros in dollars in her piggybank (let's say, about $222) and Gonzague should have the equivalent of $100 in euros in his piggybank (let's say, about 89 euros).

Special requirement of the code you have to write:
Very expectedly, you will need to define variables to hold the amount of dollars / euros that Zoe and Gonzague respectively own: please specify these variables to be of type double. So that's two double variables.
Now you might be tempted to define two new variables to hold the new amounts that each person owns. There would then be 4 double variables. In this lab activity, we do not want to allow you to do this (remember, we are concerned about memory usage and we believe we can do better than that!): you are only allowed to use 3 double variables in total in this piece of code.

Pseudocode of the solution:
We hereafter provide you with part of the pseudocode. One of the instructions for this activity will consist in completing this pseudocode.
Parts of the pseudocode that are missing are flagged as "**???**".

Note: *We will assume that the currency conversion between euros and dollars is fixed to the following: 100 euros = 111 dollars*

```
1.      Ask the user to enter the amount of dollars that Zoe owns
2.      Retrieve this amount using some kind of a listener and store it in a double
variable called Zoe
3.      Ask the user to enter the amount of euros that Gonzague owns
4.      Retrieve this amount using some kind of a listener and store it in a double
variable called Gonzague
5.      Display the original amount of dollars of Zoe
6.      Display the original amount of euros of Gonzague
7.      Create a new variable of type double, called **???**
8.      **???**
9.      **???**
10.     **???**
11.     Display the value stored in Zoe as the amount that Zoe now owns in dollars
12.     Display the value stored in Gonzague as the amount that Gonzague now owns
in euros
```

Instructions:

1. Complete the above pseudocode in the same docx file as the one you started for activity 1.
2. Complete the provided code where prompted to make sure that the code follows your pseudocode.
3. Provide, in the docx file, a command line execution of the program (see example below) for the following three scenarios:
   a. Zoe has $200 and Gonzague has 100 euros
   b. Zoe has $150 and Gonzague has 20 euros
   c. Zoe has $80.50 and Gonzague has 323.40 euros

Example of command line execution:

```
Please enter the amount of dollars in Zoe's piggy bank: 200
Please enter the amount of euros in Gonzague's piggy bank: 100
Zoe has $200 in her piggy bank.
Gonzague has 100 euros in his piggy bank.
After the trade, Zoe now owns $111 and Gonzague now owns 178 euros.
```

Do make sure that your code actually executes exactly the way it shows above. Failing to comply will sadly result in points off.

What you have to turn in:

1. The docx file with completed pseudocode and the command line executions
2. The java file with the completed program **tradingPiggyBanks**

That's it! Looking forward to seeing you in lab!