

Exercise 2: Thinking about a For Loop (on paper)
First, we'll do an in-class exercise.

Suppose that you have the following program:

```
int square = 0, sum = 0;
// for loop that sums the squares of the first 7 integers
for (int i = 1; i < 8; i++) {
    square = i * i;
    sum = sum + square;
    System.out.println("Index is: " + i + " Sum is: " + sum);
}
```

Without typing this program into NetBeans, in the following boxes, write the values of the variables as this program is executed. Note that the initial values of square and sum are 0, but that the variable i starts with the loop. For the loop values of square and sum, give the values they will have at the end of each iteration of the loop.

Variable	Initial	Val1	Val2	Val3	Val4	Val5	Val6	Val7
i								
square	0							
sum	0							

Exercise 3. Writing a Java GUI application with a For statement (at your own pace after the example discussed in class).

1. Create a new project

Go to the File -> NewProject. On the first page, leave Java and Java Application selected and on the second page, type the name of a project, for example ComputeInterest.

2. Create the GUI form and make it the main class.

In the left hand pane, right click on the name of the project (ComputeInterest) and select New -> JFrameForm. Name the form something like ComputeInterestForm.

Now again right click on the name of the project (ComputeInterest) and this time select Properties. Now select Run in the left pane, and browse to select ComputeInterestForm as the main class in the right pane.

3. Write application

In this GUI application, we assume that there is a bank offering 5% interest compounded annually on its savings accounts. On the form, the user can type in two things: their savings amount and the number of years that the amount will get interest, compounding annually. They will be shown what their savings amount would be increased to after that many years.

Make a form with

- two labels and two textfields where the user can type in their savings amount, sometimes called the balance, and the number of years,
- a button that they can click to get the interest computed,
- and a textfield or label where the new balance plus interest will be displayed. If it is a textfield, make it really wide to hold many decimal places (or format the new balance as currency).

Right click on the button and select Events -> Action -> actionPerformed to create the source code.

Add code to compute the interest in the actionPerformed method, right after the comment with TODO:

Step 1: Declare two double variables called balance and yearlyinterest and initialize them to 0.0, and an integer variable for the number of years.

```
double balance = 0.0, yearlyinterest = 0.0;  
int numyears = 0;
```

Step 2: Get the values that the user types into the textfields.

```
// get the balance (savings amount) and number of years from user  
balance = ... // add code here  
numyears = ...
```

Step 3: Write a for loop to compute the interest every year and add it to the balance to get a new balance, using the pattern worked out in class.

```
// compute 5% interest each year and add to balance
```

Step 4: And finally, display the result to the user.

```
// display the new balance to the user  
(Optional: Format the result as currency.)
```

Run this program and test it.

Try at least two sets of values for the savings amount and the number of years.

savings amount (initial balance)	number of years	new balance

To hand in this lab, print the code inside the compute interest button for the Compute Interest problem and hand it in with this sheet by Tuesday, February 4.