

## Overview

This week we are going to learn to use different loop constructs to write dynamic programs that change behavior based on the type of inputs they receive from the user. We will start with exercises on how to convert from while-loop to for-loop and do-while-loop. Then we will try to find the appropriate loop type to solve different problems.

## For-loop Syntax

General form of a for-loop is as follows.

```
for (index = initExpr; continueExpr; incrementExpr) {  
    statement 1;  
    statement 2;  
    ...  
    statement n;  
}
```

The sequence of execution follows these set of rules:

Step1: Before starting:

- Execute `index = initExpr` to set the loop index `index` to the initial value `initExpr`.

Step 2: Evaluate `continueExpr` and if it is `true` do the following:

- Execute the body statements `statement 1, ..., statement n`.
- Execute the `incrementExpr` to increment (or decrement) the loop index `index`
- Go to step 2

Loop ends when `continueExpr` is `false`.

We can put any number of statements inside the body of the loop. If it's only one statement then we don't need block `{}` curly brackets, otherwise we need them just like if-statements. Be sure to understand each of the three parts in a for-loop statement. One statement form of a for-loop is as follows:

```
for (index = initExpr; continueExpr; incrementExpr)  
    statement;
```

## Do-While-loop Syntax

One statement form of a do-while-loop is as follows.

```
do  
    statement;  
while (booleanExpr);
```

The general form of a do-while-loop has a block statement as the body:

```
do {  
    statement 1;  
    statement 2;  
    ...  
    statement n;  
} while (booleanExpr);
```

## Getting Started

After starting Eclipse, create a new project called **Lab20\_10**. Refer to Lab 9's **AllNumWhile.java** for how to write a while-loop. Import the **AllNumFor.java** and **AllNumDo.java** files from the Lab 10 assignment page into the project and load it. These programs show how the same problem can be solved in 3 different ways.

## Part 1: Create – SumAllFor.java and SumAllDo.java

Use your Lab 9 program `SumAll.java` that is written as while-loop as reference. Then create two new files and a different loop type in each. `SumAllFor.java` should contain a for-loop to implement the same logic. `SumAllDo.java` should behave similarly using a do-while-loop. Refer to your Reading 05 assignment and examples from book and lectures to note the differences and convert correctly.

## Part 2: Create – MultiplesFor.java

This program should ask the user for a maximum number to print out and then print each multiple of the base number starting from the base to the maximum. For-loop is the most appropriate for this type of program since it has an obvious starting value and ending value. Your program must produce an output as follows (inputs are shown in green):

```
This program prints out multiples of a base number till a specified maximum.
```

```
Please enter the maximum number: 25
```

```
Please enter the base number whose multiples to print: 3
```

```
Multiples of 3 from 1 till 25 are:
```

```
Number 3
```

```
Number 6
```

```
Number 9
```

```
Number 12
```

```
Number 15
```

```
Number 18
```

```
Number 21
```

```
Number 24
```

## Part 3: Create – SumSquare.java (any loop)

This program should ask the user for the maximum number to print out to and then print each number starting from 1 to the maximum along with it squared. It will also print out the sum of all the square numbers as shown below (inputs are shown in green):

```
Please enter the maximum number: 3
```

```
Number 1 squared is 1
```

```
Number 2 squared is 4
```

```
Number 3 squared is 9
```

```
The sum of squares of all numbers from 1 till 3 is: 14
```

```
-----  
Please enter the maximum number: 10
```

```
Number 1 squared is 1
```

```
Number 2 squared is 4
```

```
Number 3 squared is 9
```

```
Number 4 squared is 16
```

```
Number 5 squared is 25
```

```
Number 6 squared is 36
```

```
Number 7 squared is 49
```

```
Number 8 squared is 64
```

```
Number 9 squared is 81
```

```
Number 10 squared is 100
```

```
The sum of squares of all numbers from 1 till 10 is: 385
```

## Part 4: Create – PosAverage.java (do-while-loop)

Implement the following using a do-while-loop since this program will ask for a number at least one time. This is a prototypical reason for using a do-while-loop over the other two types of loops. This program asks the user to enter any number then do the following:

- It adds the user input to the total if the number is positive.

- Repeats to ask the user for another number.
- If the user input is not positive ( $\leq 0$ ), then the loop terminates.
- The program then outputs the average of all the positive numbers.

Your program must produce an output as follows (inputs are shown in **green**):

Enter 0 or less to stop entering numbers.

Enter value #1: 1

Enter value #2: 2

Enter value #3: 3

Enter value #4: 4

Enter value #5: 0

Average is 2.5

-----  
Enter 0 or less to stop entering numbers.

Enter value #1: -3

No positive numbers were input.

-----  
Enter 0 or less to stop entering numbers.

Enter value #1: 324

Enter value #2: 23

Enter value #3: 32

Enter value #4: 51

Enter value #5: 54

Enter value #6: 42

Enter value #7: 95

Enter value #8: 85

Enter value #9: 84

Enter value #10: 325

Enter value #11: -2

Average is 111.5

## Part 5: (Assessment) Logic Check

Create a Word document or text file named **Part5** that contains answers to the following:

1. How many iterations for each for-loop where **max = 10** and **incr = 3**
  - a. `for (int i = 0; i < max; i++)`
  - b. `for (int i = 0; i < max; i += 2)`
  - c. `for (int i = 0; i < max; i += incr)`
  - d. `for (int i = incr; i < max; i += incr)`
  - e. `for (int i = max; i > 0; i--)`
  - f. `for (int i = max; i < 0; i--)`
  - g. `for (int i = 1; i < max; i *= incr)`
  - h. `for (int i = 0; i < max; i *= incr)`
2. How will **while (true)** behave if the loop continues as long as the condition is true?
3. When should you use:
  - a. A for-loop?
  - b. A do-while-loop?

## What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

**Before** you submit, make sure you have done the following:

- Attached the file named **Part5** containing answers to the assessment questions.
- Attached the created **SumAllFor.java**, **SumAllDo.java**, **MultiplesFor.java**, **SumSquare.java** and **PosAverage.java** files.

- Filled in your collaborator's name (if any) in the "Comments..." text-box at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the end of the grace period.