

Lesson 45: Iteration Revisited #2 (W13D3)

Balboa High School

Michael Ferraro

November 12, 2015

Common deficiencies in new programmers include

- designing and implementing iterative/recursive processes to solve specific problems
- designing a structure for a program with specific results in mind; e.g., *what methods do I need aside from `main()`?*

Through **practice**, we can address these issues.

Begin flipping through the slides of this self-guided lesson, completing the programming tasks that follow. The goal is to get through as many of the tasks as you can by the end of the period with minimal assistance from others.

Aim

Students will practice writing iterative algorithms to solve problems posed to them in the lesson slides.

Setup Steps

- 1 Create a new project, Lesson45, to contain all of today's exercises.
- 2 For each exercise, create a new class. The **class name** should **match the slide title**.

Write a program that prints out the set of consecutive integers from 1 to 10, inclusive, each on a new line. Do this using a `for()` loop.

Don't move on to the following slides until you're finished!

Write a program that has a method within it that you can call from `main()` that prints out the consecutive integers from 1 to `n`, where `n` is the parameter (input value) of your method.

For example, if you include this method call in `main()` —

```
printOneToN(5);
```

— it will result in this output to the screen:

```
1
2
3
4
5
```

OneToNByM

Write a program similar to `OneToN` in which the method that does the printing takes two parameters, `n` and `m`. Instead of counting up from 1 to `n` by 1 each time, you count up by `m`.

For example, if you include this method call in `main()` —

```
printOneToNByM(32,5);
```

— it will result in this output to the screen:

```
1  
6  
11  
16  
21  
26  
31
```

CalculateInterest

Write a program that has a method you can call which takes the following parameters —

<code>double p</code>	principal balance
<code>double r</code>	annual interest rate, expressed as a decimal value; e.g., for 5%, $r = .05$
<code>int t</code>	number of years over which to accrue interest

— and returns a `double` representing the ending balance.

Call your method from `main()` like so:

```
computeInterest(15200, .039, 6);
```

Then, print the resulting amount to the screen.

Notes:

- 1 Use a `for()` loop, not an exponential function.
- 2 Assume interest is compounded only once per year.

End of Mandatory Problems

- If you have finished the problems preceding this slide and there is still class time remaining, move on to the CodingBat problems that follow.
- If you finished the preceding exercises for HW, you are not *required* to continue on to the next problems (but you are *encouraged* to!)

Extension #1: Introducing CodingBat!

From time to time, I may ask you to work on some of the problems posted at <http://codingbat.com>.¹ If you want the site to remember your solutions, sign up for a free account.

Work on these:

- [sleepIn](#)
- [diff21](#)
- [nearHundred](#)

¹Formerly known as <http://javabat.com>.

Extension #2: FindFactors

Write a program that, given an integer, will produce a printed list of all of its integer factors.

Finish the exercises in this slide set for next class.

You have a hands-on quiz coming up, so don't be surprised if you find some of these exercises rehashed as quiz problems later!