Computer Science 340
Programming Languages
Siena College
Fall 2023

Lab 2: More Python for Java Programmers

Due: 11:59 PM, Sunday, September 17, 2023

This lab is a continuation of the last, continuing your exploration of some of the features of the Python programming language, focusing on things that it does differently from Java.

This is an individual lab, but you are welcome to collaborate with classmates as you work through. However, the work you submit must ultimately be your own, and you are responsible for understanding everything in the lab.

Learning goals:

1. Learn how to use Python lists.
2. Learn how write and call functions in Python.
3. Learn how the basics of classes work in Python.

Getting Set Up

In Canvas, you will find a link to follow to set up your GitHub repository, which will be named `python2-lab-yourgitname`, for this lab.

First, A Little Math

In the first Python lab, we saw that some functions are always available to all Python programs, like `print` and `input`. This can be thought of as being analogous to Java's implicit import of the classes in the `java.lang` package. When a Python program needs to use other functions, one or more *modules* will need to be imported.

For example, to use the mathematical functions in the math module, a program should include the statement:

```
import math
```

The Connections Game asks players to take a group of 16 words and put them into 4 groups of 4, where each group is "connected" by a common theme. One can take a group of 16 items and form

$$\binom{16}{4}\cdot\binom{12}{4}\cdot\binom{8}{4}\cdot\binom{4}{4} \tag{1}$$

different groupings.

Practice Program: Write a (very short) Python program in `conn.py` that uses one or more appropriate functions from the `math` module to compute and print the number of different groupings. (5 points)

Unlike in Java, Python programmers do not need to worry about exceeding the capacity of their integer datatypes.

Practice Program: Write a Java program `Factorial.java` which computes and prints 1!, then 2!, then 3!, and so on, until the result exceeds the largest number than can be stored in a Java `long` variable. (5 points)

Practice Program: Write a Python program `factorial.py` which computes and prints 1!, then 2!, then 3!, up to 50! and notice that your answers are correct. (5 points)

Functions

You read as part of the previous lab about text and lists in Python. Now read Section 4.7 of the Python Tutorial about defining functions.

Practice Program: Write a Python function and code to test it in `func.py` that takes a list of integers as its parameter. It should construct and return a list of integers based on the input parameter subject to the following rules: (*i*) negative entries from the input are converted to the corresponding positive values, (*ii*) zero entries are omitted, and (*iii*) positive entries are doubled. So if the input `[-4 0 8 -1 0 2 -14]` is passed in, the returned list would contain `[4 16 1 4 14]` (10 points)

Practice Program: Write a Python function and code to test it in `func2.py` that takes a list of integers as its parameter. It should modify the list so that the locations of the smallest and the largest values in the input list are swapped. You should assume that the list is unsorted and contains all unique numbers. (5 points)

Classes

Section 9 of the Python Tutorial describes how classes work in Python.

Your repository contains a Java implementation of a `Ratio` class and some tests in the `main` method of the `Ratios` class. Many of you have probably seen this code before.

Practice Program: Write a Python equivalent of this program in a file `ratio.py`. You can have the "main" program in the same file rather than a separate one. (20 points)

Submission

Commit and push!

Grading

This assignment will be graded out of 50 points.

Feature	Value	Score
`conn.py`	5	
`Factorial.java`	5	
`factorial.py`	5	
`func.py`	10	
`func2.py`	5	
`ratio.py`	20	
Total	50	