



## Lab 6: Sorting Callbacks

Due: 11:59 PM, Tuesday, November 28, 2023

In this lab, we will look at the idea of callback functions. This idea can be used for things such as support of generic searching and sorting routines, or to implement iterator-like functionality over data structures.

You may work alone or in groups of size 2 or 3 on this lab.

---

### Getting Set Up

In Canvas, you will find a link to follow to set up your GitHub repository, which will be named `callbacks-lab-yourgitname`, for this lab. Only one member of the group should follow the link to set up the repository on GitHub, then others should request a link to be granted write access.

All GitHub repositories must be created with all group members having write access and all group member names specified in the `README.md` file by 12:50 PM, Tuesday, November 21, 2023. This applies to those who choose to work alone as well!

---

### Java's `Arrays.sort` Method

Recall Java's `Comparator` interface, which requires a function `compare` that is used to compare two instances of a given object type. It returns whether the first is “less than” (return value negative), “equal to” (return value 0), or “greater than” (return value positive) the second.

**Practice Program:** Write a Java program in `IntegerSorter.java` that provides the functionality listed below. (15 points)

- Creates 3 arrays of `Integer` objects containing the numbers  
4, 5, 2, -1, 0, 12  
9, -192, 2  
and  
10, 20, 30, 15, 87, -22, 928, 23, -28, -32, 9, 49152, 2048
- Sorts each array with `Arrays.sort()` using the natural ordering.
- Sorts each array in decreasing order of the number of times 2 appears in its prime factorization. You will need to write a nontrivial `Comparator` for this.

## Python Lists `sort` Method

Python lists have extensive functionality, including a `sort` method. Like Java's `Arrays.sort`, it takes an optional parameter `key=` that can be the name of a field of an object to use for the comparison value or the name of a method to call to give a numeric value to the objects that can then be used to order them.

**Practice Program:** Write a Python program `intlistsorter.py` that has the same functionality as the Java program in the previous section, except that it uses `key=` to specify the ordering for the second set of sorts. (15 points)

## C's `qsort` Function

The PDF file in the `qsort` directory of your repository is an excerpt from the classic book *The C Programming Language*, Second Edition, by Kernighan and Ritchie, Prentice Hall, 1988. That book's Section 5.11 shows you a way you can write a sort function to sort data in multiple ways by specifying a *function pointer* as one of the parameters to the sort function. This kind of comparator function works similarly to Java's `Comparator.compare` methods. In C, we achieve similar functionality but without the formal language or API support.

Make sure you understand how this works, then look at the `qsort` function provided by the C standard library (`man qsort`). This works similarly to the one in K&R, but works on an arbitrary array of any data type.

The `qsort_examples` directory has a program that shows some examples of `qsort` in action.

**Practice Program:** Write a C program `intqsorter.c` that has the same functionality as the Java and Python programs in the previous sections, except that it uses a comparator function to specify the ordering for the second set of sorts. (15 points)

## Submission

Commit and push!

## Grading

This assignment will be graded out of 45 points.

Feature	Value	Score
Java Program	15	
Python Program	15	
C Program	15	
Total	45	