Chapter 8: Recursion

Objectives

Recursion is a powerful programming technique. This chapter introduces recursive processing. It explains the basic ideas of recursion and then explores the use of recursion in programming.

Vocabulary

recursion
infinite recursion
recursion vs iteration
direct vs indirect recursion
merge sort
base case

Reading Assignment

Read pages 465 – 488 and "summary of key concepts" on page 500 in Java Software Solutions textbook.

Textbook Assignment

Book problems are due at the beginning of the class period *before* the Chapter 8 test. Problems will be checked for completion. You are encouraged to correct your answers with the solutions key during class, break or lunch.

- Self-Review Questions pg 500-501
- Multiple Choice # 8.1 – 8.8
- True False # 8.1 – 8.8
- AP Style pg 507 # 8.1 – 8.6
- AP Style Free Res. pg 511 # 8.1

PracticeIt! Assignments

Complete the following PracticeIt! assignments. For each problem, *handwrite* the solution and attach with a printed copy of your "My Problems" page *sorted with newest on top*.

<u>Chapter 12: Recursion</u>

Self-Check 12.3: mystery1
Self-Check 12.6: mysteryXY
Self-Check 12.13: mystery4
Exercise 12.1: starString
Exercise 12.8: multiplyEvens
Exercise 12.9: sumTo
Exercise 12.13: indexOf

Note: for Exercise 12.1, you will need to throw an `IllegalArgumentException` error if the value passed is less than 0. Here's an example of the syntax:

```
if ( n < 0 ) {
      throw new IllegalArgumentException("n must be greater than 0");
}
```

Labs

Download the Chapter 8 Lab Manual from GitHub. Create a `\Chapter 8` folder in your `\APCS` folder. Follow the lab manual instructions for creating the `.java` files for the labs below.

Lab	Assignment	Completed
1	Computing Powers • `Power.java`	
2	Counting and Summing Digits • `DigitPlay.java`	
3	Efficient Computation of Fibonacci Numbers • `Fib.java` • `TestFib.java`	

Fibonacci call tree for recursive call to `fib1(5)`.