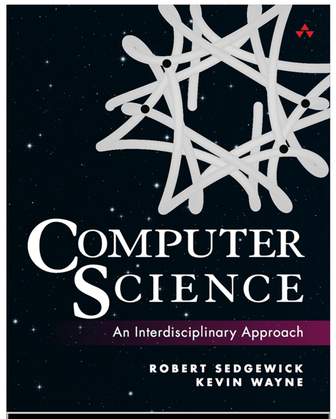

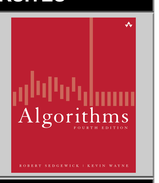

2. Functions

Overview. In this chapter, we consider a concept that has as profound an impact on control flow as do conditionals and loops: the *function*, which allows us to transfer control back and forth between different pieces of code. Functions are important because they allow us to clearly separate tasks within a program and because they provide a general mechanism that enables us to reuse code.

- *2.1 Static Methods* introduces the Java mechanism (the *static method*) for implementing functions.
- *2.2 Libraries and Clients* describes how to group related static methods into libraries to enable modular programming.
- *2.3 Recursion* considers the idea of a function calling *itself*. This possibility is known as recursion.
- *2.4 Percolation* presents a case study that uses Monte Carlo simulation to study a natural model known as percolation.

Java programs in this chapter. Below is a list of Java programs in this chapter. Click on the program name to access the Java code; click on the reference number for a brief description; read the textbook for a full discussion.

REF	PROGRAM	DESCRIPTION
2.1.1	Harmonic.java	harmonic numbers (revisited)
2.1.2	Gaussian.java	Gaussian functions
2.1.3	Coupon.java	coupon collector (revisited)
2.1.4	PlayThatTuneDeluxe.java	play that tune (revisited)
2.2.1	StdRandom.java	random number library
2.2.2	StdArrayIO.java	array I/O library
2.2.3	IFS.java	iterated function systems
2.2.4	StdStats.java	data analysis library
2.2.5	StdStats.java	data analysis library
2.2.6	Bernoulli.java	Bernoulli trials
2.3.1	Euclid.java	Euclid's algorithm
2.3.2	TowersOfHanoi.java	towers of Hanoi
2.3.3	Beckett.java	Gray code
2.3.4	Htree.java	recursive graphics
2.3.5	Brownian.java	Brownian bridge
2.3.6	LongestCommonSubsequence.java	longest common subsequence
2.4.1	Percolation.java	percolation scaffolding
2.4.2	VerticalPercolation.java	vertical percolation
2.4.3	PercolationVisualizer.java	percolation visualization client
2.4.4	PercolationProbability.java	percolation probability estimate
2.4.5	Percolation.java	percolation detection
2.4.6	PercolationPlot.java	adaptive plot client

Last modified on July 27, 2016.