

ALEXANDER A. STEPANOV
DANIEL E. ROSE



FROM
MATHEMATICS
TO
GENERIC
PROGRAMMING

From Mathematics to Generic Programming

From Mathematics to Generic Programming

Table of Contents

Cover

Title Page

Copyright Page

Contents

Acknowledgments

About the Authors

Authors' Note

1 What This Book Is About

1.1 Programming and Mathematics

1.2 A Historical Perspective

1.3 Prerequisites

1.4 Roadmap

2 The First Algorithm

2.1 Egyptian Multiplication

2.2 Improving the Algorithm

2.3 Thoughts on the Chapter

3 Ancient Greek Number Theory

3.1 Geometric Properties of Integers

3.2 Sifting Primes

3.3 Implementing and Optimizing the Code

Table of Contents

3.4 Perfect Numbers

3.5 The Pythagorean Program

3.6 A Fatal Flaw in the Program

3.7 Thoughts on the Chapter

4 Euclid's Algorithm

4.1 Athens and Alexandria

4.2 Euclid's Greatest Common Measure Algorithm

4.3 A Millennium without Mathematics

4.4 The Strange History of Zero

4.5 Remainder and Quotient Algorithms

4.6 Sharing the Code

4.7 Validating the Algorithm

4.8 Thoughts on the Chapter

5 The Emergence of Modern Number Theory

5.1 Mersenne Primes and Fermat Primes

5.2 Fermat's Little Theorem

5.3 Cancellation

5.4 Proving Fermat's Little Theorem

5.5 Euler's Theorem

5.6 Applying Modular Arithmetic

5.7 Thoughts on the Chapter

6 Abstraction in Mathematics

6.1 Groups

6.2 Monoids and Semigroups

6.3 Some Theorems about Groups

6.4 Subgroups and Cyclic Groups

Table of Contents

6.5 Lagrange's Theorem

6.6 Theories and Models

6.7 Examples of Categorical and Non-categorical
Theories

6.8 Thoughts on the Chapter

7 Deriving a Generic Algorithm

7.1 Untangling Algorithm Requirements

7.2 Requirements on A

7.3 Requirements on N

7.4 New Requirements

7.5 Turning Multiply into Power

7.6 Generalizing the Operation

7.7 Computing Fibonacci Numbers

7.8 Thoughts on the Chapter

8 More Algebraic Structures

8.1 Stevin, Polynomials, and GCD

8.2 Göttingen and German Mathematics

8.3 Noether and the Birth of Abstract Algebra

8.4 Rings

8.5 Matrix Multiplication and Semirings

8.6 Application: Social Networks and Shortest Paths

8.7 Euclidean Domains

8.8 Fields and Other Algebraic Structures

8.9 Thoughts on the Chapter

9 Organizing Mathematical Knowledge

9.1 Proofs

Table of Contents

9.2 The First Theorem

9.3 Euclid and the Axiomatic Method

9.4 Alternatives to Euclidean Geometry

9.5 Hilbert's Formalist Approach

9.6 Peano and His Axioms

9.7 Building Arithmetic

9.8 Thoughts on the Chapter

10 Fundamental Programming Concepts

10.1 Aristotle and Abstraction

10.2 Values and Types

10.3 Concepts

10.4 Iterators

10.5 Iterator Categories, Operations, and Traits

10.6 Ranges

10.7 Linear Search

10.8 Binary Search

10.9 Thoughts on the Chapter

11 Permutation Algorithms

11.1 Permutations and Transpositions

11.2 Swapping Ranges

11.3 Rotation

11.4 Using Cycles

11.5 Reverse

11.6 Space Complexity

11.7 Memory-Adaptive Algorithms

11.8 Thoughts on the Chapter

Table of Contents

12 Extensions of GCD

12.1 Hardware Constraints and a More Efficient Algorithm

12.2 Generalizing Stein's Algorithm

12.3 Bézout's Identity

12.4 Extended GCD

12.5 Applications of GCD

12.6 Thoughts on the Chapter

13 A Real-World Application

13.1 Cryptology

13.2 Primality Testing

13.3 The Miller-Rabin Test

13.4 The RSA Algorithm: How and Why It Works

13.5 Thoughts on the Chapter

14 Conclusions

Further Reading

A: Notation

B: Common Proof Techniques

B.1 Proof by Contradiction

B.2 Proof by Induction

B.3 The Pigeonhole Principle

C: C++ for Non-C++ Programmers

C.1 Template Functions

C.2 Concepts

C.3 Declaration Syntax and Typed Constants

C.4 Function Objects

Table of Contents

C.5 Preconditions, Postconditions, and Assertions

C.6 STL Algorithms and Data Structures

C.7 Iterators and Ranges

C.8 Type Aliases and Type Functions with using in
C++11

C.9 Initializer Lists in C++11

C.10 Lambda Functions in C++11

C.11 A Note about inline

Bibliography

Index