

Main

Text
Projects
Tutor
About

Related Sites

CS 61A Course
Version 1

Welcome to **Composing Programs**, a free online introduction to programming and computer science.

In the tradition of **SICP**, this text focuses on methods for abstraction, programming paradigms, and techniques for managing the complexity of large programs. These concepts are illustrated primarily using the **Python 3** programming language.

In addition to reading the chapters below, you can apply your knowledge to the **programming projects** that accompany the text and visualize program execution using the **Online Python Tutor**.

Instructors: If you are interested in adapting any of these materials for your courses, please **fill out this short survey** so that we can support your efforts.

Chapter 1: Building Abstractions with Functions

- 1.1 Getting Started
- 1.2 Elements of Programming
- 1.3 Defining New Functions
- 1.4 Designing Functions
- 1.5 Control
- 1.6 Higher-Order Functions
- 1.7 Recursive Functions

Chapter 2: Building Abstractions with Data

- 2.1 Introduction
- 2.2 Data Abstraction
- 2.3 Sequences
- 2.4 Mutable Data
- 2.5 Object-Oriented Programming
- 2.6 Implementing Classes and Objects
- 2.7 Object Abstraction
- 2.8 Efficiency
- 2.9 Recursive Objects

Chapter 3: Interpreting Computer Programs

- 3.1 Introduction
- 3.2 Functional Programming
- 3.3 Exceptions
- 3.4 Interpreters for Languages with Combination
- 3.5 Interpreters for Languages with Abstraction

Chapter 4: Data Processing

- 4.1 Introduction
- 4.2 Implicit Sequences
- 4.3 Declarative Programming
- 4.4 Logic Programming
- 4.5 Unification
- 4.6 Distributed Computing
- 4.7 Distributed Data Processing
- 4.8 Parallel Computing

Composing Programs by John DeNero, based on the textbook Structure and Interpretation of Computer Programs by Harold Abelson and Gerald Jay Sussman, is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.