

Design Concepts in Programming Languages

FRANKLYN TURBAK AND
DAVID GIFFORD
WITH MARK A. SHELDON

Hundreds of programming languages are in use today—scripting languages for Internet commerce, user interface programming tools, spreadsheet macros, page format specification languages, and many others. Designing a programming language is a metaprogramming activity that bears certain similarities to programming in a regular language, with clarity and simplicity even more important than in ordinary programming. This comprehensive text uses a simple and concise framework to teach key ideas in programming language design and implementation. The book’s unique approach is based on a family of syntactically simple pedagogical languages that allow students to explore programming language concepts systematically. It takes as its premise and starting point the idea that when language behaviors become incredibly complex, the description of the behaviors must be incredibly simple.

The book presents a set of tools (a mathematical metalanguage, abstract syntax, operational and denotational semantics) and uses it to explore a comprehensive set of programming language design dimensions, including dynamic semantics (naming, state, control, data), static semantics (types, type reconstruction, polymorphism, effects), and pragmatics (compilation, garbage collection). The many examples and exercises offer students opportunities to apply the foundational ideas explained in the text. Specialized topics and code that implements many of the algorithms and compilation methods in the book can be found on the book’s Web site, along with such additional material as a section on concurrency and proofs of the theorems in the text. The book is suitable as a text for an introductory graduate or advanced undergraduate programming languages course; it can also serve as a reference for researchers and practitioners.

COVER PHOTOGRAPH: DAVID GIFFORD

On the cover is an inuksuk, a signpost used by the Inuit in the Arctic to provide guidance in vast wilderness. The concise semantics, type rules, effect rules, and compilation transforms in this book have been valuable inuksuit to the authors in the programming language landscape.

THE MIT PRESS MASSACHUSETTS INSTITUTE OF TECHNOLOGY CAMBRIDGE, MASSACHUSETTS 02142 [HTTP://MITPRESS.MIT.EDU](http://mitpress.mit.edu)

Franklyn Turbak is Associate Professor in the Computer Science Department at Wellesley College. David Gifford is Professor of Computer Science and Engineering at MIT. Mark A. Sheldon is Visiting Assistant Professor in the Computer Science Department at Wellesley College.

“There is a paucity of good graduate-level textbooks on the foundations of programming languages, no more than four or five in the last two decades. Nothing to compare with the profusion of excellent texts in the other core areas of computer science, such as algorithms or operating systems. This new textbook by Franklyn Turbak, David Gifford, and Mark Sheldon—comprehensive, thorough, pedagogically innovative, impeccably written and organized—greatly enriches the area of programming languages and will be an important reference for years to come.”

Assaf Kfoury
Department of Computer Science, Boston University

“This book is an excellent, systematic exploration of ideas and techniques in programming language theory. The book carefully, but without wasting time on extraneous complications, explains operational and denotational semantic techniques and their application to many aspects of programming language design. It will be of great value for graduate courses and for independent study.”

Gary T. Leavens
School of Electrical Engineering and Computer Science,
University of Central Florida



TURBAK, GIFFORD,
AND SHELDON
Design Concepts
in Programming Languages



Design Concepts in Programming Languages

FRANKLYN TURBAK AND
DAVID GIFFORD
WITH MARK A. SHELDON

