

1. Record Nr.	UNINA9910427669403321
Autore	Jones Cliff B.
Titolo	Understanding programming languages / / Cliff B. Jones
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2020] Â©2020
ISBN	3-030-59257-X
Edizione	[1st ed. 2020.]
Descrizione fisica	1 online resource (XI, 229 p. 42 illus., 17 illus. in color.)
Disciplina	005.13
Soggetti	Programming languages (Electronic computers) Computer programming
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Programming Languages -- Delimiting a Language -- Operational Semantics -- Typing -- Block Structure -- Additional Issues in Sequential Languages -- Alternative Semantic Approaches -- Shared Variable Concurrency -- Concurrent OOLs -- Exceptional Ordering -- Conclusions -- App. A, Simple Language -- App. B, Typed Language -- App. C, Blocks Language -- App. D, COOL -- App. E, VDM Notation.
Sommario/riassunto	This book is about describing the meaning of programming languages. The author teaches the skill of writing semantic descriptions as an efficient way to understand the features of a language. While a compiler or an interpreter offers a form of formal description of a language, it is not something that can be used as a basis for reasoning about that language nor can it serve as a definition of a programming language itself since this must allow a range of implementations. By writing a formal semantics of a language a designer can yield a far shorter description and tease out, analyse and record design choices. Early in the book the author introduces a simple notation, a meta-language, used to record descriptions of the semantics of languages. In a practical approach, he considers dozens of issues that arise in current programming languages and the key techniques that must be mastered in order to write the required formal semantic descriptions. The book concludes with a discussion of the eight key challenges: delimiting a language (concrete representation), delimiting the abstract content of a

language, recording semantics (deterministic languages), operational semantics (non-determinism), context dependency, modelling sharing, modelling concurrency, and modelling exits. The content is class-tested and suitable for final-year undergraduate and postgraduate courses. It is also suitable for any designer who wants to understand languages at a deep level. Most chapters offer projects, some of these quite advanced exercises that ask for complete descriptions of languages, and the book is supported throughout with pointers to further reading and resources. As a prerequisite the reader should know at least one imperative high-level language and have some knowledge of discrete mathematics notation for logic and set theory.
