Programming With Objects A Comparative Presentation Of Object Oriented Programming With C And Java

Learn how to write object-oriented programs in R and how to construct classes and class hierarchies in the three object-oriented systems available in R. This book gives an introduction to object-oriented programming in the R programming language and shows you how to use and apply R in an object-oriented manner. You will then be able to use this powerful programming style in your own statistical programming projects to write flexible and extendable software. After reading Advanced Object-Oriented Programming in R, you'll come away with a practical project that you can reuse in your own analytics coding endeavors. You'll then be able to visualize your data as objects that have state and then manipulate those objects with polymorphic or generic methods. Your projects will benefit from the high degree of flexibility provided by polymorphism, where the choice of concrete method to execute depends on the type of data being manipulated. What You'll Learn Define and use classes and generic functions using R Work with the R class hierarchies Benefit from implementation reuse Handle operator overloading Apply the S4 and R6 classes Who This Book Is For Experienced programmers and for those with at least some prior experience with R programming language. /div
As execution speeds reach the physical limits of single cpu computers, the only hope of achieving greater computing power is with parallel systems. Researchers have proposed countless new programming languages, but their differences, similarities, strengths, weaknesses and problem domains are subtle and often not well understood. Informed comparison of parallel languages is difficult. This volume compares eight parallel programming languages based on solutions to four problems. Each chapter includes a description of the language's philosophy, semantics and syntax, and a solution to each problem. By considering solutions rather than language features or theoretical properties, the gap is bridged between the language specialists and users. Both professionals and students in the fields of computer and computational science will find the discussions helpful and understandable.
Object-Oriented scripting with Perl and Python Scripting languages are becoming increasingly important for software development. These higher-level languages, with their built-in easy-to-use data structures are convenient for programmers to use as "glue" languages for assembling multi-language applications and for quick prototyping of software architectures. Scripting languages are also used extensively in Web-based applications. Based on the same overall philosophy that made Programming with Objects such a wide success, Scripting with Objects takes a novel dual-language approach to learning advanced scripting with Perl and Python, the dominant languages of the genre. This method of comparing basic syntax and writing application-level scripts is designed to give readers a more comprehensive and expansive perspective on the subject. Beginning with an overview of the importance of scripting languages—and how they differ from mainstream systems programming languages—the book explores: Regular expressions for string processing The notion of a class in Perl and Python Inheritance and polymorphism in Perl and Python Handling exceptions Abstract classes and methods in Perl and Python Weak references for memory management Scripting for graphical user interfaces Multithreaded scripting Scripting for network programming Interacting with databases Processing XML with Perl and Python This book serves as an excellent textbook for a one-semester undergraduate course on advanced scripting in which the students have some prior experience using Perl and Python, or for a two-semester course for students who will be experiencing scripting for the first time. Scripting with Objects is also an ideal resource for industry professionals who are making the transition from Perl to Python, or vice versa.
A Comparative Study of Programming Languages
Scripting with Objects

Generalizing the Programming Function
A Comparative Presentation of Object-Oriented Programming With C++ and Java
In programming courses, using the different syntax of multiple languages, such as C++, Java, PHP, and Python, for the same abstraction often confuses students new to computer science. Introduction to Programming Languages separates programming language concepts from the restraints of multiple language syntax by discussing the concepts at an abstrac
C++ is a general purpose programming language that, in addition to systems applications, is extensively used for scientific computation, financial applications, embedded systems, realtime control, and other applications. Emphasizing the commonality between C++ and Java as object oriented languages, this text prepares the reader to program with objects.
Abstract: "The paper evaluates Oberon-2, Modula-3, Sather, and Self in the context of object-oriented programming. While each of these programming languages provide support for classes with inheritance, dynamic dispatch, code reuse, and information hiding, they do so in very different ways and with varying levels of efficiency and simplicity. A single application was coded in each language and the experience gained forms the foundation on which the subjective critique is based. By comparing the actual run-times of the various implementations it is also possible to present an objective analysis of the efficiency of the languages. Furthermore, by coding the application using both explicit dynamic dispatch and static method binding, it is possible to evaluate the cost of dynamic dispatch in each language. The application was also coded in C++, thereby providing a well know [sic] baseline against which the execution times can be compared."
Programming with Objects
DAIMI PB.
A Comparative Presentation of Object-Oriented Scripting with Perl and Python
A Comparative Study of Five Parallel Programming Languages
Comparative semantics of programming languages / [by] Brian H. Mayoh
This guide was written for readers interested in learning the C++ programming language from scratch, and for both novice and advanced C++ programmers wishing to enhance their knowledge of C++. The text is organized to guide the reader from elementary language concepts to professional software development, with in depth coverage of all the C++ language elements en route.
Write code that's clean, concise, and to the point: code that others will read with pleasure and reuse. Comparing your code to that of expert programmers is a great way to improve your coding skills. Get hands-on advice to level up your coding style through small and understandable examples that compare flawed code to an improved solution. Discover handy tips and tricks, as well as common bugs an experienced Java programmer needs to know. Make your way from a Java novice to a master craftsman. This book is a useful companion for anyone learning to write clean Java code. The authors introduce you to the fundamentals of becoming a software craftsman, by comparing pieces of problematic code with an improved version, to help you to develop a sense for clean code. This unique before-and-after approach teaches you to create clean Java code. Learn to keep your booleans in check, dodge formatting bugs, get rid of magic numbers, and use the right style of iteration. Write informative comments when needed, but avoid them when they are not. Improve the understandability of your code for others by following conventions and naming your objects accurately. Make your programs more robust with intelligent exception handling and learn to assert that everything works as expected using JUnit5 as your testing framework. Impress your peers with an elegant functional programming style and clear-cut object-oriented class design. Writing excellent code isn't just about implementing the functionality. It's about the small important details that make your code more readable, maintainable, flexible, robust, and faster. Java by Comparison teaches you to spot these details and trains you to become a better programmer. What You Need: You need a Java 8 compiler, a text editor, and a fresh mind.That's it.
Programming Languages: Concepts and Implementation teaches language concepts from two complementary perspectives: implementation and paradigms. It covers the implementation of concepts through the incremental construction of a progressive series of interpreters in Python, and Racket Scheme, for purposes of its combined simplicity and power, and assessing the differences in the resulting languages. 1. Hands-on, implementation-oriented approach. 2. Numerous conceptual and programming exercises. 3. Interpreter-based projects in Python and Racket Scheme. 4. All interpreter code (and solutions) in Python (and Racket) are provided as a Git repository in BitBucket. 5. New concurrency models (Communicating Sequential Processes (CSP), and Actor Model of Concurrency).
Comparison of an Object-oriented Programming Language to a Procedural Programming Language for Effectiveness in Program Maintenance
Become a Java Craftsman in 80 Examples
Nondeterminism and Recursion
A Comparative Study on Procedural Type Object Oriented Programming Languages
Comparative Metric Semantics of Programming Languages
A Comparative Presentation of Object-Oriented Programming with C++ and Java, a comparative presentation of object-oriented programming with two of the most popular programming languages of today, teaches vital skills and techniques for the Internet age. Based on highly successful courses taught by the author, this book answers the need for a comprehensive educational program on the subject of object-oriented programming. In a clear and accessible format, the author compares and contrasts both languages, from basic language constructs to how both languages are used in application-level programming, such as graphics programming, network programming, and database programming. Learning to write a program in one language that corresponds to a given program in the other language enables students to tackle more difficult projects in either language.
To non-specialists in the field, the phrase "a programming language" is usually held to mean "one of those things like Autocode, Fortran, Algol or Cobol, which are supposed to make programming language easier."
This comprehensive examination of the main approaches to object-oriented language explains key features of the languages in use today. Class-based, prototypes and Actor languages are all examined and compared in terms of their semantic concepts. This book provides a unique overview of the main approaches to object-oriented languages. Exercises of varying length, some of which can be extended into mini-projects are included at the end of each chapter. This book can be used as part of courses on Comparative Programming Languages or Programming Language Semantics at Second or Third Year Undergraduate Level. Some understanding of programming language concepts is required.
20th Brazilian Symposium, SBLP 2016, Maringá, Brazil, September 22-23, 2016, Proceedings
Java by Comparison
Raspberry Pi :Raspberry Pi Guide On Python & Projects Programming In Easy Steps
Concepts Of Programming Languages
A Thesis
Covers the nature of language, syntax, modeling objects, names, expressions, functions, control structures, global control, logic programming, representation and semantics of types, modules, generics, and domains
Scripting with ObjectsA Comparative Presentation of Object-Oriented Scripting with Perl and PythonJohn Wiley & Sons
This reference is intended for experienced practitioners, consultants and students working on building practical applications. It discusses the most widely-used programming languages and their fuctional pros and cons for application and development. The author provides: a brief overview of programming languages principles and concepts; numerous diagrams, charts and sample programs; coverage of object-oriented programming and visual programming; and tables rating languages on such subjects as simplicity, data structuring, portability and efficiency.
A Complete Guide to Programming in C++
A Comparison of Object-oriented Programming in Four Modern Languages
Object-Oriented Programming and Java
A Comparative Presentation of Object Oriented Programming with C++
Statistical Programming for Data Science, Analysis and Finance
Covering the latest in Java technologies, Object-Oriented Programming and Java teaches the subject in a systematic, fundamentals-first approach. It begins with the description of real-world object interaction scenarios and explains how they can be translated, represented and executed using object-oriented programming paradigm. By establishing a solid foundation in the understanding of object-oriented programming concepts and their applications, this book provides readers with the pre-requisites for writing proper object-oriented programs using Java.
bull; Demonstrates how Python is the perfect language for text-processing functions. bull; Provides practical pointers and tips that emphasize efficient, flexible, and maintainable approaches to text-processing challenges. bull; Helps programmers develop solutions for dealing with the increasing amounts of data with which we are all inundated.
This book constitutes the proceedings of the 20th Brazilian Symposium on Progamming Languages, SBLP 2016, held in Maringá, Brazil, in September 2016. The 12 papers presented in this volume were carefully reviewed and selected from 26 submissions. They deal with fundamental principles and innovations in the design and implementation of programming languages and systems.
Object-Oriented Programming Languages: Interpretation
Overview and Comparison
Comparative semantics of programming languages
An Experiment in Comparative Programming Languages
Programming Languages
A text for a comparative language course (as well as for practicing computer programmers), considering the principal programming language concepts and showing how they are dealt with in traditional imperative languages, such as Pascal, C, and Ada, in functional languages such as ML, in logic languages like PROLOG, in purely object-oriented language.
"Raspberry Pi Programming Guide" is a text that gives the reader a bit of insight into this form of technology. It is European based and is just making a debut in North America so many are curious about it and what exactly this technology can do. The aim that the author has with this text is to highlight the main functions of Raspberry Pi and how it can be beneficial to the consumer in the long run. The text is extremely informative and to the point and it is simple to read. The great thing about the book is that anyone, even someone who does not know much about this form of technology can understand the process. It is a great text to have in any household that has a keen interest in technology.
During the last three decades several different styles of semantics for program ming languages have been developed. This book compares two of them: the operational and the denotational approach. On the basis of several exam ples we show how to define operational and denotational semantic models for programming languages. Furthermore, we introduce a general technique for comparing various semantic models for a given language. We focus on different degrees of nondeterminism in programming lan guages. Nondeterminism arises naturally in concurrent languages. It is also an important concept in specification languages. In the examples discussed, the degree of non determinism ranges from a choice between two alternatives to a choice between a collection of alternatives indexed by a closed interval of the real numbers. The former arises in a language with nondeterministic choices. A real time language with dense choices gives rise to the latter. We also consider the nondeterministic random assignment and parallel composition, both couched in a simple language. Besides non determinism our four example languages contain some form of recursion, a key ingredient of programming languages.
A Comparison of Modern Object-oriented Programming Languages and Compilers Using Standard Gang of Four Design Patterns
Text Processing in Python
Comparative Programming Languages
A Comparative Introduction
Comparative Semantics of Programming Languages
Abstract: "We will take up the Smalltalk-80, the Simula and the Delta as typical programming languages from procedural type object oriented programming languages. The purpose of this study is to compare these three languages about the following aspects: how to represent an object, program structure, reference method for attributes of objects and redefinition of attributes, forms of procedure representation, multiple inheritance and concurrent programming support."
Advanced Object-Oriented Programming in R
A Comparative Study of Three Object-oriented Programming Environments
Comparative Study of the Semantics of Select Programming Languages
A Comparative Study of Object-oriented and Conventional Programming
A Comparative Study of Parallel Programming Languages: The Salishan Problems