

Inside The C Object Model

Introduction to Programming with C++ for Engineers
 Object-Oriented Design and Programming with C++
 Object-Oriented Systems in C++
 Data Clustering in C++
 Advanced R
 Programming with Objects
 C# For Java Programmers
 Design Patterns in Modern C++
 C++ Object Databases
 Document Object Model
 Learning DCOM
 Advanced C++ Programming Cookbook
 Inside the Object Model
 C++ and Object-oriented Programming
 Applying UML and Patterns
 Practical Statecharts in C/C++
 Programming in C++
 Object Oriented Programming in C++
 OBJECT-ORIENTED PROGRAMMING USING C++
 Object-Oriented Programming in C++
 Class Construction in C and C++
 Python Programming On Win32
 Object-Oriented Neural Networks in C++
 Inside the Object Model
 Essential COM
 Python in a Nutshell
 An Introduction to Object-Oriented Programming in C++
 Game Programming Patterns
 Understanding the DOM — Document Object Model
 Object-Oriented Computation in C++ and Java
 ECOOP '91 European Conference on Object-Oriented Programming
 Real-Time C++
 Theoretical Aspects of Object-oriented Programming
 Object-Oriented Graphics Programming in C++
 Structured and Object-oriented Problem Solving Using C++
 Inside COM
 Mastering Object-Oriented Design in C++
 Deciphering Object-Oriented Programming with C++
 Inside the C++ Object Model
 Inside the C++ Object Model

Inside The C Object Model

Downloaded from music-school.fhny.org by guest

TREVON HEATH

Introduction to Programming with C++ for Engineers Wiley-IEEE Press

The book presents an up-to-date overview of C++ programming with object-oriented programming concepts, with a wide coverage of classes, objects, inheritance, constructors, and polymorphism. Selection statements, looping, arrays, strings, function sorting and searching algorithms are discussed. With abundant practical examples, the book is an essential reference for researchers, students, and professionals in programming.

Object-Oriented Design and Programming with C++ Firewall Media

This is the digital version of the printed book (Copyright 2007). Virtually all business, scientific, and engineering applications are heavily reliant on numeric data items. C++ and Java offer object-oriented programmers unique flexibility and control over the computations required within such applications. However, most books on object-oriented programming gloss over such numeric data

items, emphasizing instead one-dimensional containers or collections and components of the graphical user interface. Object-Oriented Computation in C++ and Java fills the gap left by such books. Drawing on more than twenty years' experience as a software developer, tester, consultant, and professor, Conrad Weisert shows readers how to use numeric objects effectively. Not limited to any language or methodology, the concepts and techniques discussed in this book are entirely independent of one's choice of design and coding methodology. Practitioners of Extreme Programming, UML-driven design, agile methods, incremental development, and so on will all develop these same data classes. Whether you are a seasoned professional or an advanced computer science student, this book can teach you techniques that will improve the quality of your programming and the efficiency of your applications. The exercises (and answers) presented in this book will teach you new ways to implement the computational power of C++, Java, and numeric data items. Topics include taxonomy of data types developing and using object-oriented classes for numeric data design patterns for commonly occurring numeric data types families of interacting numeric data types choosing efficient and flexible internal data representations

techniques for exploiting pattern reuse in C++ conventions for arithmetic operations in Java numeric vectors and matrices

Object-Oriented Systems in C++ MIT Press

Here is the ultimate guide to creating and extending documents within the application programming interface of the Document Object Model (DOM). The book examines real-world applications of the DOM, including exclusive case studies of DOM-based browsers and applications and provides a comprehensive, language-neutral examination of the DOM and its related applications.

Data Clustering in C++ Packt Publishing Ltd

Object Oriented Programming in C++ Object Oriented Programming is a programming in which we design and develop our application or program based of object. Objects are instances(variables) of class.Object oriented programming does not allow data to flow freely around the system. It binds data more closely to the functions that operate on it, and protects it from accidental modifications from outside functions.Object oriented programming allows separation of a complex programs into

objects and then builds data and functions around these objects. The data of an object can be accessed only by the functions associated with that object. However, functions of one object can access the functions of other objects. Features of OOP's (Object Oriented Programming) Class: Class is an encapsulation of data and coding. Classes are an expanded version of structures. Structure can contain multiple variables. Classes can contain multiple variables, even more, classes can also contain functions as class member. Variables available in class are called Data Members. Functions available in class are called Member Functions. Object: Class is a user-defined data type and object is a variable of class type. Object is used to access class members. Inheritance: Inheritance means access the properties and features of one class into another class. The class who is going to provide its features to another class will be called base class and the class who is using the properties and features of another class will be called derived class. Polymorphism: Polymorphism means more than one function with same name, with different working. It can be static or dynamic. In static polymorphism memory will be allocated at compile time. In dynamic polymorphism memory will be allocated at runtime. Both function overloading and operator overloading are an examples of static polymorphism. Virtual function is an example of dynamic polymorphism. Data Abstraction: The basic idea of data abstraction is to visible only the necessary information, unnecessary information will be hidden from the outside world. This can be done by making class members as private members of class. Private members can be accessed only within the same class where they are declared. Encapsulation: Encapsulation is a process of wrapping data members and member functions in a single unit called class. Using the method of encapsulation, the programmer cannot directly access the data. Data is only accessible through the object of the class.

Advanced R Addison-Wesley Professional

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.

Programming with Objects Springer Science & Business Media

There is a lot of misinformation and myth about the overhead and costs associated with C++. Now Stan Lippman, the acclaimed author of the C++ Primer, answers the call for a book that gives strategy guidelines for C++ programming. Inside the C++ Object Model explains where overhead costs reside and what they actually consist of. The author explains which parts vary by implementation and which are invariant. He tells how the various implementation models arose, points out areas where they are likely to evolve, and explains why they are what they are. This book is a must for C++ programmers who want to understand the semantic implications of the C++ object model and how the model affects their programs.

C# For Java Programmers "O'Reilly Media, Inc."

Apply modern C++17 to the implementations of classic design patterns. As well as covering traditional design patterns, this book fleshes out new patterns and approaches that will be useful to C++ developers. The author presents concepts as a fun investigation of how problems can be solved in different ways, along the way using varying degrees of technical sophistication and explaining different sorts of trade-offs. Design Patterns in Modern C++ also provides a technology demo for modern C++, showcasing how some of its latest features (e.g., coroutines) make difficult problems a lot easier to solve. The examples in this book are all suitable for putting into production, with only a few simplifications made in order to aid readability. What You Will Learn Apply design patterns to modern C++ programming Use creational patterns of builder, factories, prototype and singleton Implement structural patterns such as adapter, bridge, decorator, facade and more Work with the behavioral patterns such as chain of responsibility, command, iterator, mediator and more Apply functional design patterns such as Monad and more Who This Book Is For Those with at least some prior programming experience, especially in C++.

Design Patterns in Modern C++ DigitalOcean

Data clustering is a highly interdisciplinary field, the goal of which is to divide a set of objects into homogeneous groups such that objects in the same group are similar and objects in different groups are quite distinct. Thousands of theoretical papers and a number of books on data clustering have been published over the past 50 years. However,

C++ Object Databases Walter de Gruyter GmbH & Co KG

"This book is distinctive in that it implements nodes and links as base objects and then composes them into four different kinds of neural networks. Roger's writing is clear....The text and code are

both quite readable. Overall, this book will be useful to anyone who wants to implement neural networks in C++ (and, to a lesser extent, in other object-oriented programming languages)...I recommend this book to anyone who wants to implement neural networks in C++."--D.L. Chester, Newark, Delaware in COMPUTING REVIEWSOBJECT-ORIENTED NEURAL NETWORKS IN C++ IS A VALUABLE TOOL FOR ANYONE WHO WANTS TO UNDERSTAND, IMPLEMENT, OR UTILIZE NEURAL NETWORKS. THIS BOOK/DISK PACKAGE PROVIDES THE READER WITH A FOUNDATION FROM WHICH ANY NEURAL NETWORK ARCHITECTURE CAN BE CONSTRUCTED. THE AUTHOR HAS EMPLOYED OBJECT-ORIENTED DESIGN AND OBJECT-ORIENTED PROGRAMMING CONCEPTS TO DEVELOP A SET OF FOUNDATION NEURAL NETWORK CLASSES, AND SHOWS HOW THESE CLASSES CAN BE USED TO IMPLEMENT A VARIETY OF NEURAL NETWORK ARCHITECTURES WITH A GREAT DEAL OF EASE AND FLEXIBILITY. A WEALTH OF NEURAL NETWORK FORMULAS (WITH STANDARDIZED NOTATION), OBJECT CODE IMPLEMENTATIONS, AND EXAMPLES ARE PROVIDED TO DEMONSTRATE THE OBJECT-ORIENTED APPROACH TO NEURAL NETWORK ARCHITECTURES AND TO FACILITATE THE DEVELOPMENT OF NEW NEURAL NETWORK ARCHITECTURES. THIS IS THE FIRST BOOK TO TAKE FULL ADVANTAGE OF THE REUSABLE NATURE OF NEURAL NETWORK CLASSES. KEY FEATURES * DESCRIBES HOW TO USE THE CLASSES PROVIDED TO IMPLEMENT A VARIETY OF NEURAL NETWORK ARCHITECTURES INCLUDING ADALINE, BACKPROPAGATION, SELF-ORGANIZING, AND BAM * PROVIDES A SET OF REUSABLE NEURAL NETWORK CLASSES, CREATED IN C++, CAPABLE OF IMPLEMENTING ANY NEURAL NETWORK ARCHITECTURE * INCLUDES AN IBM DISK OF THE SOURCE CODE FOR THE CLASSES, WHICH IS PLATFORM INDEPENDENT * INCLUDES AN IBM DISK WITH C++ PROGRAMS DESCRIBED IN THE BOOK

Document Object Model McGraw Hill Professional

Offers a discussion of all the advanced and object-oriented features of C++. Hands-on examples show how features are used in real programming situations. Contains a coding style guide that shows users how to program more effectively and enables them to gain experience with professional style guides. Chapter two provides a crash course which is accessible to programmers in any procedural language.

Learning DCOM Academic Press

Object-Oriented Programming in C++ begins with the basic principles of the C++ programming language and systematically introduces increasingly advanced topics while illustrating the OOP methodology. While the structure of this book is similar to that of the previous edition, each chapter reflects the latest ANSI C++ standard and the examples have been thoroughly revised to reflect current practices and standards. Educational Supplement Suggested solutions to the programming projects found at the end of each chapter are made available to instructors at recognized educational institutions. This educational supplement can be found at www.prenhall.com, in the Instructor Resource Center.

Advanced C++ Programming Cookbook CRC Press

With this book, Christopher Kormanyos delivers a highly practical guide to programming real-time embedded microcontroller systems in C++. It is divided into three parts plus several appendices. Part I provides a foundation for real-time C++ by covering language technologies, including object-oriented methods, template programming and optimization. Next, part II presents detailed descriptions of a variety of C++ components that are widely used in microcontroller programming. It details some of C++'s most powerful language elements, such as class types, templates and the STL, to develop components for microcontroller register access, low-level drivers, custom memory management, embedded containers, multitasking, etc. Finally, part III describes mathematical methods and generic utilities that can be employed to solve recurring problems in real-time C++. The appendices include a brief C++ language tutorial, information on the real-time C++ development environment and instructions for building GNU GCC cross-compilers and a microcontroller circuit. For this third edition, the most recent specification of C++17 in ISO/IEC 14882:2017 is used throughout the text. Several sections on new C++17 functionality have been added, and various others reworked to reflect changes in the standard. Also several new sample projects are introduced and existing ones extended, and various user suggestions have been incorporated. To facilitate portability, no libraries other than those specified in the language standard itself are used. Efficiency is always in focus and numerous examples are backed up with real-time performance measurements and size analyses that quantify the true costs of the code down to the very last byte and microsecond. The target audience of this book mainly consists of students and professionals interested in real-time C++. Readers should be familiar with C or another programming language and will benefit most if they have had some previous experience with microcontroller electronics and the performance and size issues prevalent in embedded systems programming.

Inside the Object Model CRC Press

ECOOP '91 is the fifth annual European Conference on Object-Oriented Programming. From their beginning, the ECOOP conferences have been very successful as a forum of high scientific quality where the newest developments connected to object-oriented programming and related areas could be presented and discussed. Over the last few years object-oriented technology has gained widespread use and considerable popularity. In parallel with this, the field has matured scientifically, but there is still a lot of room for new ideas and for hot debates over fundamental issues, as these proceedings show. The 22 papers in this volume were selected by the programme committee from 129 submissions. Important issues discussed in the contributions are language design, specification, databases, concurrency types and software development.

C++ and Object-oriented Programming "O'Reilly Media, Inc."

A recipe-based guide to refining your C++ programming skills with the help of coding best practices, advanced programming concepts, and the latest features of C++17 and C++20 Key Features Learn how to develop and design your own libraries Find solutions to your app development problems and implement them in a highly reusable manner, following library development best practices Explore advanced C++ features such as containers, coroutines, and modules Book Description If you think you've mastered C++ and know everything it takes to write robust applications, you'll be in for a surprise. With this book, you'll gain comprehensive insights into C++, covering exclusive tips and interesting techniques to enhance your app development process. You'll kick off with the basic principles of library design and development, which will help you understand how to write reusable and maintainable code. You'll then discover the importance of exception safety, and how you can avoid unexpected errors or bugs in your code. The book will take you through the modern elements of C++, such as move semantics, type deductions, and coroutines. As you advance, you'll delve into template programming - the standard tool for most library developers looking to achieve high code reusability. You'll explore the STL and learn how to avoid common pitfalls while implementing templates. Later, you'll learn about the problems of multithreaded programming such as data races, deadlocks, and thread starvation. You'll also learn high-performance programming by using benchmarking tools and libraries. Finally, you'll discover advanced techniques for debugging and testing to ensure code reliability. By the end of this book, you'll have become an expert at C++ programming and will have gained the skills to solve complex development problems with ease. What you will learn Solve common C++ development problems by implementing solutions in a more generic and reusable way Achieve different levels of exception safety guarantees by introducing precise declarations Write library-quality code that meets professional standards Practice writing reliable, performant code that exposes consistent behavior in programs Understand why you need to implement design patterns and how it's done Work with complex examples to understand various aspects of good library design Who this book is for This book is for intermediate and expert-level C++ developers who are looking to explore the lesser known functionalities of the language to improve the efficiency of their code and the way they develop applications. Basic knowledge of object-oriented programming concepts and the Standard Template Library (STL) is assumed.

Applying UML and Patterns John Wiley & Sons

Object-Oriented Design and Programming with C++: Your Hands-On Guide to C++ Programming, with Special Emphasis on Design, Testing, and Reuse provides a list of software engineering principles to guide the software development process. This book presents the fundamentals of the C++ language. Organized into two parts encompassing 10 chapters, this book begins with an overview of C++ and describes object-oriented programming and the history of C++. This text then introduces classes, polymorphism, inheritance, and overloading. Other chapters consider the C++ preprocessor and organization of class libraries. This book discusses as well the scope rules, separate compilation, class libraries, and their organization, exceptions, browsers, and exception handling. The final chapter deals with the design of a moderately complex system that provides file system stimulation. This book is a valuable resource for readers who are reasonably familiar with the C programming language and want to understand the issues in object-oriented programming using C++.

Practical Statecharts in C/C++ "O'Reilly Media, Inc."

DCOM -- the Distributed Component Object Model -- is a recent upgrade of a time-honored and well-tested technology promoted by Microsoft for distributed object programming. Now that components are playing a larger and larger part in Windows 98, Windows NT 4.0, and Windows 2000, every Windows programmer will want to understand the technology. DCOM competes with CORBA as a rich and robust method for creating expandable and flexible components, allowing you

to plug in new parts conveniently and upgrade without the need for code changes to every program that uses your component. This book introduces C++ programmers to DCOM and gives them the basic tools they need to write secure, maintainable programs. While using Visual C++ development tools and wizards where appropriate, the author never leaves the results up to magic. The C++ code used to create distributed components and the communications exchanged between systems and objects are described at a level where the reader understands their significance and can use the insights for such tasks as debugging and improving performance. The first few chapters explain both the remote procedure calls that underlie DCOM's communication and the way DCOM uses C++ classes. Readers become firmly grounded in the relation between components, classes, and objects, the ways objects are created and destroyed, how clients find servers, and the basics of security and threading. After giving you a grounding in how DCOM works, this book introduces you to the Microsoft tools that make it all easy. By showing what really happens each time you choose a button in a wizard, Learning DCOM makes it possible for you to choose what you need. This book is for anyone who wants to understand DCOM. While thoroughly practical in its goals, it doesn't stint on the background you need to make your programs safe, efficient, and easy to maintain. Topics include: MIDL (Microsoft Interface Definition Language, the language for defining COM interfaces) COM error and exception handling Custom, dispatch, and

dual interfaces Standard and custom factories Management of in-process versus out-of-process servers Distributed memory management Pragmatic explanation of the DCOM wire protocol Standard, custom, handler, and automation marshaling Multithreading and apartments Security at the system configuration and programming level Active Template Library (ATL), ATL wizards -- and what they don't do Writing a component that can be invoked from Visual Basic Techniques for using distributed components Creating an ActiveX control and embedding it in a Web client Authentication and the use of Windows NT security features Techniques for merging marshaling code Connection and distributed events management An introduction to COM+ features

Programming in C++ Wiley
 'Downright revolutionary... the title is a major understatement... 'Quantum Programming' may ultimately change the way embedded software is designed.' -- Michael Barr, Editor-in-Chief, Embedded Systems Programming magazine (Click here)

Object Oriented Programming in C++ Addison-Wesley Professional
 This book teaches object-oriented analysis and design from first principles and clearly explains C++ mechanisms that implement object-oriented concepts.

OBJECT-ORIENTED PROGRAMMING USING C++ CRC Press
 The biggest challenge facing many game programmers is completing their game. Most game projects fizzle out, overwhelmed by the complexity of their own code. Game Programming Patterns

tackles that exact problem. Based on years of experience in shipped AAA titles, this book collects proven patterns to untangle and optimize your game, organized as independent recipes so you can pick just the patterns you need. You will learn how to write a robust game loop, how to organize your entities using components, and take advantage of the CPU's cache to improve your performance. You'll dive deep into how scripting engines encode behavior, how quadtrees and other spatial partitions optimize your engine, and how other classic design patterns can be used in games.

Object-Oriented Programming in C++ Addison-Wesley Professional
 Appropriate for Introductory Computer Science (CS1) courses using C++ and Introductory C++ programming courses found in Computer Science, Engineering, CIS, MIS, and Business Departments. This accessible text emphasizes problem-solving techniques using the C++ language, with coverage that develops strong problem-solving skills using problem abstraction and stepwise refinement through the "Programmer's Algorithm." Staugaard first emphasizes the structured (procedural) paradigm, then gradually advances to the object-oriented paradigm using object-oriented programming "seed topics." This approach prepares students for in-depth coverage of classes and objects presented later in the text, while building essential structured programming concepts.