Object Oriented Application Development Using Java

Learning Object-Oriented Programming is an easy-to-follow guide full of hands-on examples of solutions to common problems with object-oriented code in Python, JavaScript, and C#. It starts by helping you to recognize objects from real-life scenarios and demonstrates that working with them makes it simpler to write code that is easy to understand and reuse. You will learn to protect and hide data with the data encapsulation features of Python, JavaScript, and C#. You will explore how to maximize code reuse by writing code capable of working with objects of different types, and discover the advantage of duck typing in both Python and JavaScript, while you work with interfaces and generics in C#. With a fair understanding of interfaces, multiple inheritance, and composition, you will move on to refactor existing code and to organize your source for easy maintenance and extension. Learning Object-Oriented Programming will help you to make better, stronger, and reusable code.

Develop the next killer Android App using Java programming! Android is everywhere! It runs more than half the smartphones in the U.S.—and Java makes it go. If you want to cash in on its popularity

by learning to build Android apps with Java, all the easy-to-follow guidance you need to get started is at your fingertips. Inside, you'll learn the basics of Java and grasp how it works with Android; then, you'll go on to create your first real, working application. How cool is that? The demand for Android apps isn't showing any signs of slowing, but if you're a mobile developer who wants to get in on the action, it's vital that you get the necessary Java background to be a success. With the help of Java Programming for Android Developers For Dummies, you'll quickly and painlessly discover the ins and outs of using Java to create groundbreaking Android apps—no prior knowledge or experience required! Get the know-how to create an Android program from the ground up Make sense of basic Java development concepts and techniques Develop the skills to handle programming challenges Find out how to debug your app Don't sit back and watch other developers release apps that bring in the bucks! Everything you need to create that next killer Android app is just a page away!

This text covers what students need to know about basic Java programming in a clear, straight-forward writing style. Important Notice: Media content referenced within the product description or the product text may not be available in the ebook version.

Unleash the true power of JavaScript by mastering

Object-Oriented programming principles and patterns About This Book Covering all the new Object-Oriented features introduced in ES6, this book shows you how to build large-scale web apps Build apps that promote scalability, maintainability, and reusability Learn popular Object-Oriented programming (OOP) principles and design patterns to build robust apps Implement Object-Oriented concepts in a wide range of front-end architectures Who This Book Is For This book is ideal for you if you are a JavaScript developers who wants to gain expertise in OOP with JavaScript to improve your web development skills and build professional quality web applications. What You Will Learn Master JavaScript's OOP features, including the one's provided by ES6 specification Identify and apply the most common design patterns such as Singleton, Factory, Observer, Model-View-Controller, and Mediator Patterns Understand the SOLID principles and their benefits Use the acquired OOP knowledge to build robust and maintainable code Design applications using a modular architecture based on SOLID principles In Detail ECMAScript 6 introduces several new Object-Oriented features that drastically change the way developers structure their projects. Web developers now have some advanced OOP functionality at their disposal to build large-scale applications in JavaScript. With this book, we'll provide you with a comprehensive overview of OOP

principles in JavaScript and how they can be implemented to build sophisticated web applications. Kicking off with a subtle refresher on objects, we'll show you how easy it is to define objects with the new ES6 classes. From there, we'll fly you through some essential OOP principles, forming a base for you to get hands-on with encapsulation. You'll get to work with the different methods of inheritance and we'll show you how to avoid using inheritance with Duck Typing. From there, we'll move on to some advanced patterns for object creation and you'll get a strong idea of how to use interesting patterns to present data to users and to bind data. We'll use the famous promises to work with asynchronous processes and will give you some tips on how to organize your code effectively. You'll find out how to create robust code using SOLID principles and finally, we'll show you how to clearly define the goals of your application architecture to get better, smarter, and more effective coding. This book is your one-way ticket to becoming a JavaScript Jedi who can be counted on to deliver flexible and maintainable code. Style and approach This comprehensive guide on advanced OOP principles and patterns in JavaScript is packed with real-world use cases, and shows you how to implement advanced OOP features to build sophisticated web applications that promote scalability and reusability.

This comprehensive guide, developed at IBM's

International Technical Support Center in San Jose, CA, is ideal for managers and developers looking to apply object-oriented methods in large-scale information technology environments. Demonstrating the ups, downs, and trade-offs of object-oriented methodologies, the authors provide a wealth of information that will help managers make choices about the resources and technologies available for application development. The book discusses the impact of object technology on management decisions with examples from real, full-scale environments in which productivity has increased from the use of this technology. Features of Object Technology: provides a solid explanation of the principles of object-oriented technology; describes and compares the methodologies in use in object-oriented development environments; analyzes the role of CASE tools in object-oriented development; presents the issues involved in building a user interface; includes a full chapter on the process of team building in object-oriented applications; details the use of a configuration management tool; and explains how "legacy code" can be reused in object-oriented environments.

Three years ago, in response to our challenging development context, the Advanced Modeling and Analysis Section designed and implemented an object-oriented environment -- the Application Interface Engine (AIE). Our prototyping requirements

forced existing application development systems beyond their capabilities. Programmers at AMAS and its contractors have developed over twenty applications using AIE. Our initial experience has been very positive. AIE extends an object-oriented programming language with syntax and classes to support applications specification. This extended system improves all stages of the application engineering life cycle, from rapid prototyping to long term maintenance.

Get complete coverage of the Object-Oriented development process! Utilizing Java to develop object-oriented information systems, this text moves from an overview of key concepts and Java fundamentals into a building case study utilizing UML models.

Nowadays, newly developed software is often already obsolete by the time it is introduced. The object-oriented concept provides a solution to this "crisis," by allowing objects to be used in a wide range of programs. Object-oriented applications development with databases places special demands on the DBMS and the development environment. This book provides a detailed description of the object model of the Cach post-relational database. In addition, the reader is guided step-by-step through the development of a post-relational application. The accompanying CD-ROM contains the associated Windows software.

Nowadays, newly developed software packages are often obsolete already at the time of their introduction.

Object-oriented software development is a possible—if not the only—solution to this dilemma: applications are modeled as software objects that describe the properties and the behavior of real-world entities. Such objects are encapsulated, in that they hide—behind a publicly known interface—the complexity of their internal data structures and behaviors. This enables objects to be used in a wide range of program packages without needing to know the details of their internal implementation. Linking object-oriented modeled applications with a database places special demands on a database management system and development environment when the usual performance and semantics losses are to be avoided. This book provides a detailed description of the object model of the Caché postrelational database. This second, revised and expanded edition includes the many new features of Caché 5. There is a comprehensive description of the new Caché Studio with its improvements for developing and debugging applications as well as a whole new chapter about XML and SOAP based Web Services. The chapters about Java, ActiveX and the SQL manager have undergone a complete revision.

This practical book tells readers how to actually build object-oriented models using UML notation, and how to implement these models using Java. The authors introduce all of the basic fundamentals necessary to start applying and understanding the object-oriented paradigm without having to be an expert in computer science or advanced mathematics. It can help the reader to make the right decisions to meet their individual business

needs. Using cases, recommended approach scenarios, and examples, this clearly-written book covers a multitude of topics: managing complexity, principles of Object-Orientation, specification models, current techniques, behaviors, relationships, rules, design, Java background and fundamentals, multi-tasking, JAR files, security, Swing Applets, class and interface, internationalization, and implementing generalization and specialization. For professional software analysts and developers who work on large systems, and others in the field of computer science.

Presents an introduction to Objective-C, covering such topics as classes and objects, data types, program looping, inheritance, polymorphism, variables, memory management, and archiving.

Break free from procedural programming and learn how to optimize your applications and enhance your skills using objects and design patterns.

"This book explores applications and approaches to object-oriented software design"--

'Programming .NET Components', second edition, updated to cover .NET 2.0., introduces the Microsoft .NET Framework for building components on Windows platforms. From its many lessons, tips, and guidelines, readers will learn how to use the .NET Framework to program reusable, maintainable, and robust components.

Object-oriented programming (OOP) has been the leading paradigm for developing software applications for at least 20 years. Many different methodologies, approaches, and techniques have

been created for OOP, such as UML, Unified Process, design patterns, and eXtreme Programming. Yet, the actual process of building good software, particularly large, interactive, and long-lived software, is still emerging. Software engineers familiar with the current crop of methodologies are left wondering, how does all of this fit together for designing and building software in real projects? This handbook from one of the world's leading software architects and his team of software engineers presents guidelines on how to develop high-quality software in an application-oriented way. It answers questions such as: * How do we analyze an application domain utilizing the knowledge and experience of the users? * What is the proper software architecture for large, distributed interactive systems that can utilize UML and design patterns? * Where and how should we utilize the techniques and methods of the Unified Process and eXtreme Programming? This book brings together the best of research, development, and day-to-day project work. "The strength of the book is that it focuses on the transition from design to implementation in addition to its overall vision about software development." -Bent Bruun Kristensen, University of Southern Denmark, Odense

Connecting with students of all levels in the Introductory Programming course, Gary Bronson utlizes the groundbreaking features of JDK 5.0 in this

Enhanced Edition of his successful Java text.
Create scalable, reusable high-quality JavaScript applications and libraries
The Complete Guide to Writing More Maintainable, Manageable, Pleasing, and Powerful Ruby Applications Ruby's widely admired ease of use has a downside: Too many Ruby and Rails applications have been created without concern for their long-term maintenance or evolution. The Web is awash in Ruby code that is now virtually impossible to change or extend. This text helps you solve that problem by using powerful real-world object-oriented design techniques, which it thoroughly explains using simple and practical Ruby examples. Sandi Metz has distilled a lifetime of conversations and presentations about object-oriented design into a set of Ruby-focused practices for crafting manageable, extensible, and pleasing code. She shows you how to build new applications that can survive success and repair existing applications that have become impossible to change. Each technique is illustrated with extended examples, all downloadable from the companion Web site,poodr.info. The first title to focus squarely on object-oriented Ruby application design, Practical Object-Oriented Design in Ruby will guide you to superior outcomes, whatever your previous Ruby experience. Novice Ruby programmers will find specific rules to live by; intermediate Ruby programmers will find valuable

principles they can flexibly interpret and apply; and advanced Ruby programmers will find a common language they can use to lead development and guide their colleagues. This guide will help you Understand how object-oriented programming can help you craft Ruby code that is easier to maintain and upgrade Decide what belongs in a single Ruby class Avoid entangling objects that should be kept separate Define flexible interfaces among objects Reduce programming overhead costs with duck typing Successfully apply inheritance Build objects via composition Design cost-effective tests Solve common problems associated with poorly designed Ruby code

SNMP++ is Hewlett-Packard's new API for simplifying SNMP applications development using C++ and object-oriented techniques. In this book, you'll learn how SNMP++ will help you build network management applications that are more portable, powerful, extensible, and reliable. And if that's not enough, you'll also discover how SNMP++ can help you get to market faster than ever before. Written by a Hewlett-Packard engineer who helped develop SNMP++, this is the first book to cover the entire SNMP product development lifecycle, from analysis through design and implementation. Step-by-step, you'll learn how to use SNMP++ to build powerful applications with a minimum of coding effort.

This volume presents an introduction to Visual Basic

.NET. Visual Basic .NET (VB.NET) is an object-oriented computer programming language. It is an extension of the BASIC programming language that combines BASIC functions and commands with visual controls. Visual Basic provides a graphical user interface GUI that allows the developer to drag and drop objects into the program as well as manually write program code. Visual Basic is designed to make software development easy and efficient, while still being powerful enough to create advanced programs. For example, the Visual Basic language is designed to be "human readable," which means the source code can be understood without requiring lots of comments. Programs created with Visual Basic can be designed to run on Windows, on the Web, within Office applications, or on mobile devices. Visual Studio .NET provides development tools to create programs based on the .NET framework, such as ASP.NET applications, which are often deployed on the Web.

Introduces CRC (Class, Responsibility, Collaborator) cards and describes how they can be used in interactive sessions to develop an object-oriented model of an application.

Jia (software engineering, DePaul University) helps readers develop skills in designing software, and especially in writing object- oriented programs using Java. The text provides broad coverage of object-oriented technology, including object-oriented modeling

using the Unified Modeling Language (UML), object-oriented design using design patterns, and object-oriented programming using Java. This second edition offers expanded coverage of design patterns, enhanced material on UML, and a new introduction to the iterative software development process made popular by extreme programming. Learning features include chapter summaries, exercises, and projects.

For senior/graduate level courses on Object Oriented Design using C++, and the Booch (BC) - OOD book. A practical, problem-solving approach to the fundamental concepts of Object Oriented Design and their application using C++. This book is written for the "engineer in the trenches". It is a serious guide for practitioners of Object-Oriented design. The style is narrative, and accessible for the beginner, and yet the topics are covered in enough depth to be relevant to the consumate designer. The principles of OOD explained, one by one, and then demonstrated with numerous examples and case studies.

The Object-Oriented Thought Process Third Edition Matt Weisfeld An introduction to object-oriented concepts for developers looking to master modern application practices. Object-oriented programming (OOP) is the foundation of modern programming languages, including C++, Java, C#, and Visual Basic .NET. By designing with objects rather than treating the code and data as separate entities, OOP allows objects to fully utilize other objects' services as well as inherit their functionality. OOP promotes code portability and reuse, but requires a shift in thinking to be fully understood. Before jumping

into the world of object-oriented programming languages, you must first master The Object-Oriented Thought Process. Written by a developer for developers who want to make the leap to object-oriented technologies as well as managers who simply want to understand what they are managing, The Object-Oriented Thought Process provides a solution-oriented approach to object-oriented programming. Readers will learn to understand object-oriented design with inheritance or composition, object aggregation and association, and the difference between interfaces and implementations. Readers will also become more efficient and better thinkers in terms of object-oriented development. This revised edition focuses on interoperability across various technologies, primarily using XML as the communication mechanism. A more detailed focus is placed on how business objects operate over networks, including client/server architectures and web services. “Programmers who aim to create high quality software–as all programmers should–must learn the varied subtleties of the familiar yet not so familiar beasts called objects and classes. Doing so entails careful study of books such as Matt Weisfeld’s The Object-Oriented Thought Process.” –Bill McCarty, author of Java Distributed Objects, and Object-Oriented Design in Java Matt Weisfeld is an associate professor in business and technology at Cuyahoga Community College in Cleveland, Ohio. He has more than 20 years of experience as a professional software developer, project manager, and corporate trainer using C++, Smalltalk, .NET, and Java. He holds a BS in systems analysis, an MS in computer science, and an

MBA in project management. Weisfeld has published many articles in major computer trade magazines and professional journals.

With this book, software engineers, project managers, and tool builders will be able to better understand the role of analysis and design in the object-oriented (OO) software development process. This book presents a minimum set of notions and shows the reader how to use these notions for OO software construction. The emphasis is on development principles and implementation.

Written for programmers familiar with Java, this guide explains the principles of object-oriented programming, and how to translate object-oriented designs into real programs using Java and the unified modeling language (UML). Separate chapters address the development of graphical user interfaces with the Swing library, design patterns, and refactoring. The CD-ROM contains a personal edition of Borland's JBuilder 5. Annotation copyrighted by Book News Inc., Portland, OR.

Practical OO development tips for the C++ and Java programmer Practical Object-Oriented Development in C++ and Java offers advice on real-world ways to use these powerful programming languages and techniques. Using the Unified Modeling Language (UML) methodology, expert Cay S. Horstmann gives you clear, concise explanations of object-oriented design, C++, and Java in a way that makes these potentially daunting operations more accessible than they've ever been before. Horstmann compares and contrasts features of C++ and Java to give you a deeper understanding of OO

design. He separates the genuinely useful C++, Java, and UML features from the less effective and potentially harmful ones. Horstmann shows you how to determine the best programming practice for whatever application you're in; provides the kind of eye-opening design tips and style rules that can only come from experience; and demystifies advanced topics like frameworks and object persistence. Dozens of illuminating programming examples are readily accessible through the accompanying Web site. Useful code is available for smart pointers, easy output formatting in C++ and Java, a set of classes that makes STL safe to use, and a nifty utility that automatically extracts header files. This unique book: * Offers over 100 practical design hints for good class design * Covers the essential OO features of Java 1.1-like serialization and reflection * Uses the C++ Standard Template Library (STL) throughout * Covers CRC cards in addition to UML

Case studies implemented in several object-oriented programming languages including CÊÊ, Smalltalk, Objective-C, Actor and Object pascal.

An Essential Reference for Intermediate and Advanced R Programmers Advanced R presents useful tools and techniques for attacking many types of R programming problems, helping you avoid mistakes and dead ends. With more than ten years of experience programming in R, the author illustrates the elegance, beauty, and flexibility at the heart of R. The book develops the necessary skills to produce quality code that can be used in a variety of circumstances. You will learn: The fundamentals of R, including standard data types and functions Functional programming as a useful framework for solving wide classes of problems The positives

and negatives of metaprogramming How to write fast, memory-efficient code This book not only helps current R users become R programmers but also shows existing programmers what's special about R. Intermediate R programmers can dive deeper into R and learn new strategies for solving diverse problems while programmers from other languages can learn the details of R and understand why R works the way it does.

The acclaimed beginner's book on object technology now presents UML 2.0, Agile Modeling, and the latest in object development techniques.

Discover the untapped features of object-oriented programming and use it with other software tools to code fast, efficient applications. Key Features Explore the complexities of object-oriented programming (OOP) Discover what OOP can do for you Learn to use the key tools and software engineering practices to support your own programming needs Book Description Your experience and knowledge always influence the approach you take and the tools you use to write your programs. With a sound understanding of how to approach your goal and what software paradigms to use, you can create high-performing applications quickly and efficiently. In this two-part book, you'll discover the untapped features of object-oriented programming and use it with other software tools to code fast and efficient applications. The first part of the book begins with a discussion on how OOP is used today and moves on to analyze the ideas and problems that OOP doesn't address. It continues by deconstructing the complexity of OOP, showing you its fundamentally simple core. You'll see that, by using the distinctive elements of OOP, you can learn to build your applications more easily. The next part of this book talks about acquiring the skills to become a better programmer. You'll get an overview of how various tools, such as version control and build management,

help make your life easier. This book also discusses the pros and cons of other programming paradigms, such as aspect-oriented programming and functional programming, and helps to select the correct approach for your projects. It ends by talking about the philosophy behind designing software and what it means to be a "good" developer. By the end of this two-part book, you will have learned that OOP is not always complex, and you will know how you can evolve into a better programmer by learning about ethics, teamwork, and documentation. What you will learn Untangle the complexity of object-oriented programming by breaking it down to its essential building blocks Realize the full potential of OOP to design efficient, maintainable programs Utilize coding best practices, including TDD, pair programming and code reviews, to improve your work Use tools, such as source control and IDEs, to work more efficiently Learn how to most productively work with other developers Build your own software development philosophy Who this book is for This book is ideal for programmers who want to understand the philosophy behind creating software and what it means to be “good” at designing software. Programmers who want to deconstruct the OOP paradigm and see how it can be reconstructed in a clear, straightforward way will also find this book useful. To understand the ideas expressed in this book, you must be an experienced programmer who wants to evolve their practice.

This book describes in detail how ARIS methods model and identify business processes by means of the UML (Unified Modeling Language), leading to an information model that serves as the basis for a systematic and intelligent development of application systems. Multiple real-world examples using SAP R/3 illustrate aspects of business process modeling including methods of knowledge management, implementation of workflow systems and

standard software solutions, and the deployment of ARIS methods.

A practical tutorial showing how to use Smalltalk-80 to construct object-oriented software applications. Hopkins offers numerous worked examples and sample code explaining how to implement programs in Smalltalk-80 language. Also treats the Model-View Controller (MVC) paradigm.

Object-Oriented Programming under Windows presents object-oriented programming (OOP) techniques that can be used in Windows programming. The book is comprised of 15 chapters that tackle an area in OOP. Chapter 1 provides an introductory discourse about OOP, and Chapter 2 covers the programming languages. Chapter 3 deals with the Windows environment, while Chapter 4 discusses the creation of application. Windows and dialogue boxes, as well as controls and standard controls, are tackled. The book then covers menus and event response. Graphics operation, clipboard, bitmaps, icons, and cursors are also dealt with. The book also tackles disk file access, and then discusses the help file system. The last chapter covers data transfer. The text will be of great use to individuals who want to write Windows based programs.

Ayer helps application programmers keep up, walking them through the entire process of object-oriented, client/server application development. The book describes techniques for the analysis, design, development and implementation of a C/S-based application. Ayer thoroughly covers all OOP systems design hardware and network issues. Includes case studies and diagrams throughout.