
Effective Tcl Tk Programming Writing Better Programs In Tcl And Tk

Conversations with the Creators of Major Programming Languages

CGI Programming with Tcl

Advanced CORBA® Programming with C++

Tcl/Tk

Beginning Linux?Programming

Advanced UNIX Programming

A Complete Introduction to the Python Language

The Pragmatic Programmers' Guide

Effective STL

TCP/IP Illustrated

Tcl and the Tk Toolkit

Writing Better Programs with Tcl and Tk

The Art of UNIX Programming

Tcl/Tk 8.5 Programming Cookbook

Programming in Lua

Learn to create modern GUIs using Tkinter by building real-world projects in Python

Writing Better Programs with Tcl and Tk

Effective C++

An Investigation on the Applicability of Inter-disciplinary Concepts to Software System Development

Programming in Python 3

The Pragmatic Programmer

Effective Tcl/Tk Programming

From Journeyman to Master

Managing Corporate Information Systems Evolution and Maintenance

Elements of Reusable Object-Oriented Software (Adobe Reader)

Tcl and the Tk Toolkit

Python Projects

Design Patterns

A Developer's Guide

Avoiding Common Problems in Coding and Design

Open Sources

Develop responsive and powerful GUI applications with Tkinter

50 Specific Ways to Improve Your Use of the Standard Template Library

A Desktop Quick Reference

Effective Tcl/Tk Programming

Tcl/Tk in a Nutshell

Building Secure Software

How to Avoid Security Problems the Right Way

C++ Gotchas

Cybernetics Oriented Programming (CYBOP)

*Effective Tcl Tk Programming Writing Better Programs In
Tcl And Tk*

Downloaded from blog.gmercyyu.edu by guest

ROWE HEATH

Conversations with the Creators of Major Programming Languages "O'Reilly Media, Inc."

A guide to completing Python projects for those ready to take their skills to the next level Python Projects is the ultimate resource for the Python programmer with basic skills who is ready to move beyond tutorials and start building projects. The preeminent guide to bridge the gap between learning and doing, this book walks readers through the "where" and "how" of real-world Python programming with practical, actionable instruction. With a focus on real-world functionality, Python Projects details the ways that Python can be used to complete daily tasks and bring efficiency to businesses and individuals alike. Python Projects is written specifically for those who know the Python syntax and lay of the land, but may still be intimidated by larger, more complex projects. The book provides a walk-through of the basic set-up for an application and the building and packaging for a library, and explains in detail the functionalities related to the projects. Topics include: *How to maximize the power of the standard library modules *Where to get third party libraries, and the best practices for utilization *Creating, packaging, and reusing libraries within and across projects *Building multi-layered functionality including networks, data, and user interfaces *Setting up development environments and using virtualenv, pip, and more Written by veteran Python trainers, the book is structured for easy navigation and logical progression that makes it ideal for individual, classroom, or corporate training. For Python developers looking to apply their skills to real-world challenges, Python Projects is a goldmine of information and expert insight.

CGI Programming with Tcl John Wiley & Sons

The Art of UNIX Programming poses the belief that understanding the unwritten UNIX engineering tradition and mastering its design patterns will help programmers of all stripes to become better programmers. This book attempts to capture the engineering wisdom and design philosophy of the UNIX, Linux, and Open Source software development community as it has evolved over the past three decades, and as it is applied today by the most experienced programmers. Eric Raymond offers the next generation of "hackers" the unique opportunity to learn the connection between UNIX philosophy and practice through careful case studies of the very best UNIX/Linux programs.

Advanced CORBA® Programming with C++ "O'Reilly Media, Inc."

A tutorial and reference to the object-oriented programming language for beginning to experienced programmers, updated for version 1.8, describes the language's structure, syntax, and operation, and explains how to build applications. Original. (Intermediate)

Tcl/Tk Packt Publishing Ltd

Describes the concepts of programming with Linux, covering such topics as shell programming, file structure, managing memory, using MySQL, debugging, processes and signals, and GNOME.

[Beginning Linux?Programming](#) Addison-Wesley Professional

The Tcl Programming Language is a comprehensive guide to the current version (8.6) of this immensely flexible and versatile language. Starting with the basic features, it expands its scope to include the more advanced concepts, facilities and programming idioms from which the language derives its power. Begin with the basics of Tcl syntax and commands for operating on data. Get acquainted with the flexible and uniform execution model that enables metaprogramming, custom control structures etc. Learn to modularize your code with namespaces, object-oriented design and packages. See how intrinsic support for Unicode and encodings makes it a breeze to localize your applications. Become conversant with the integrated event loop and how it facilitates efficient asynchronous I/O models and the reactive style of programming. Delve into Tcl's sophisticated I/O framework and write your own reflected channels, transforms and virtual file systems. Understand the built-in facilities for inter-process communication using pipes or the network. See how concurrent programming facilities like coroutines and threads can simplify your code and make it more performant. Learn how to secure your application through the use of safe interpreters for sandboxing. Interact with databases through the Tcl Database Connectivity interface. Discover how software distribution and installation headaches are eliminated with starkits and single file deployment. The breadth of coverage and numerous examples will familiarize newcomers to every aspect of Tcl programming. At the same time, the depth and level of detail, and an exhaustive index, make The Tcl Programming Language a valuable reference in every Tcl programmer's library.

Advanced UNIX Programming Manning Publications

Scripting with Python makes you productive and increases the reliability of your scientific work. Here, the author teaches you how to develop tailored, flexible, and efficient working environments built from small programs (scripts) written in Python. The focus is on examples and applications of relevance to computational science: gluing existing applications and tools, e.g. for automating simulation, data analysis, and visualization; steering simulations and computational experiments; equipping programs with graphical user interfaces; making computational Web services; creating interactive interfaces with a Maple/Matlab-like syntax to numerical applications in C/C++ or Fortran; and building flexible object-oriented programming interfaces to existing C/C++ or Fortran libraries.

A Complete Introduction to the Python Language Addison-Wesley Professional

Masterminds of Programming features exclusive interviews with the creators of several historic and highly influential programming languages. In this unique collection, you'll learn about the processes that led to specific design decisions, including the goals they had in mind, the trade-offs they had to make, and how their experiences have left an impact on programming today. Masterminds of Programming includes individual interviews with: Adin D. Falkoff: APL Thomas E. Kurtz: BASIC Charles H. Moore: FORTH Robin Milner: ML Donald D. Chamberlin: SQL Alfred Aho, Peter Weinberger, and Brian Kernighan: AWK Charles Geschke and John Warnock: PostScript Bjarne Stroustrup: C++ Bertrand Meyer: Eiffel Brad Cox and Tom Love: Objective-C Larry Wall: Perl Simon Peyton Jones, Paul Hudak, Philip Wadler, and John Hughes: Haskell Guido van Rossum: Python Luiz Henrique de Figueiredo and Roberto Ierusalimsky: Lua James Gosling: Java Grady Booch, Ivar Jacobson, and

James Rumbaugh: UML Anders Hejlsberg: Delphi inventor and lead developer of C# If you're interested in the people whose vision and hard work helped shape the computer industry, you'll find *Masterminds of Programming* fascinating.

The Pragmatic Programmers' Guide Pearson Education

Freely available source code, with contributions from thousands of programmers around the world: this is the spirit of the software revolution known as Open Source. Open Source has grabbed the computer industry's attention. Netscape has opened the source code to Mozilla; IBM supports Apache; major database vendors have ported their products to Linux. As enterprises realize the power of the open-source development model, Open Source is becoming a viable mainstream alternative to commercial software. Now in *Open Sources*, leaders of Open Source come together for the first time to discuss the new vision of the software industry they have created. The essays in this volume offer insight into how the Open Source movement works, why it succeeds, and where it is going. For programmers who have labored on open-source projects, *Open Sources* is the new gospel: a powerful vision from the movement's spiritual leaders. For businesses integrating open-source software into their enterprise, *Open Sources* reveals the mysteries of how open development builds better software, and how businesses can leverage freely available software for a competitive business advantage. The contributors here have been the leaders in the open-source arena: Brian Behlendorf (Apache) Kirk McKusick (Berkeley Unix) Tim O'Reilly (Publisher, O'Reilly & Associates) Bruce Perens (Debian Project, Open Source Initiative) Tom Paquin and Jim Hamerly (mozilla.org, Netscape) Eric Raymond (Open Source Initiative) Richard Stallman (GNU, Free Software Foundation, Emacs) Michael Tiemann (Cygnus Solutions) Linus Torvalds (Linux) Paul Vixie (Bind) Larry Wall (Perl) This book explains why the majority of the Internet's servers use open-source technologies for everything from the operating system to Web serving and email. Key technology products developed with open-source software have overtaken and surpassed the commercial efforts of billion dollar companies like Microsoft and IBM to dominate software markets. Learn the inside story of what led Netscape to decide to release its source code using the open-source mode. Learn how Cygnus Solutions builds the world's best compilers by sharing the source code. Learn why venture capitalists are eagerly watching Red Hat Software, a company that gives its key product -- Linux -- away. For the first time in print, this book presents the story of the open-source phenomenon told by the people who created this movement. *Open Sources* will bring you into the world of free software and show you the revolution.

Effective STL "O'Reilly Media, Inc."

Most organizations have a firewall, antivirus software, and intrusion detection systems, all of which are intended to keep attackers out. So why is computer security a bigger problem today than ever before? The answer is simple--bad software lies at the heart of all computer security problems. Traditional solutions simply treat the symptoms, not the problem, and usually do so in a reactive way. This book teaches you how to take a proactive approach to computer security. *Building Secure Software* cuts to the heart of computer security to help you get security right the first time. If you are serious about computer security, you need to read this book, which includes essential lessons for both security professionals who have come to realize that software is the problem, and software developers who intend to make their code behave. Written for anyone involved in software

development and use—from managers to coders—this book is your first step toward building more secure software. *Building Secure Software* provides expert perspectives and techniques to help you ensure the security of essential software. If you consider threats and vulnerabilities early in the development cycle you can build security into your system. With this book you will learn how to determine an acceptable level of risk, develop security tests, and plug security holes before software is even shipped. Inside you'll find the ten guiding principles for software security, as well as detailed coverage of: Software risk management for security Selecting technologies to make your code more secure Security implications of open source and proprietary software How to audit software The dreaded buffer overflow Access control and password authentication Random number generation Applying cryptography Trust management and input Client-side security Dealing with firewalls Only by building secure software can you defend yourself against security breaches and gain the confidence that comes with knowing you won't have to play the "penetrate and patch" game anymore. Get it right the first time. Let these expert authors show you how to properly design your system; save time, money, and credibility; and preserve your customers' trust.

TCP/IP Illustrated IGI Global

Python is rapidly becoming the de facto standard language for systems integration. Python has a large user and developer base external to the neuroscience community, and a vast module library that facilitates rapid and maintainable development of complex and intricate systems. In this Research Topic, we highlight recent efforts to develop Python modules for the domain of neuroscience software and neuroinformatics: - simulators and simulator interfaces - data collection and analysis - sharing, re-use, storage and databasing of models and data - stimulus generation - parameter search and optimization - visualization - VLSI hardware interfacing. Moreover, we seek to provide a representative overview of existing mature Python modules for neuroscience and neuroinformatics, to demonstrate a critical mass and show that Python is an appropriate choice of interpreter interface for future neuroscience software development.

Tcl and the Tk Toolkit "O'Reilly Media, Inc."

"This is *Effective C++* volume three - it's really that good." - Herb Sutter, independent consultant and secretary of the ISO/ANSI C++ standards committee "There are very few books which all C++ programmers must have. Add *Effective STL* to that list." - Thomas Becker, Senior Software Engineer, Zephyr Associates, Inc., and columnist, *C/C++ Users Journal* C++'s Standard Template Library is revolutionary, but learning to use it well has always been a challenge. Until now. In this book, best-selling author Scott Meyers (*Effective C++* , and *More Effective C++*) reveals the critical rules of thumb employed by the experts - the things they almost always do or almost always avoid doing - to get the most out of the library. Other books describe what's in the STL. *Effective STL* shows you how to use it. Each of the book's 50 guidelines is backed by Meyers' legendary analysis and incisive examples, so you'll learn not only what to do, but also when to do it - and why. Highlights of *Effective STL* include: Advice on choosing among standard STL containers (like vector and list), nonstandard STL containers (like hash_set and hash_map), and non-STL containers (like bitset). Techniques to maximize the efficiency of the STL and the programs that use it. Insights into the behavior of iterators, function objects, and allocators, including things you should not do. Guidance for the proper use of algorithms and member functions whose names are the same (e.g., find), but

whose actions differ in subtle (but important) ways. Discussions of potential portability problems, including straightforward ways to avoid them. Like Meyers' previous books, *Effective STL* is filled with proven wisdom that comes only from experience. Its clear, concise, penetrating style makes it an essential resource for every STL programmer.

Writing Better Programs with Tcl and Tk Pearson Education

TCP/IP Illustrated, an ongoing series covering the many facets of TCP/IP, brings a highly-effective visual approach to learning about this networking protocol suite. *TCP/IP Illustrated, Volume 2* contains a thorough explanation of how TCP/IP protocols are implemented. There isn't a more practical or up-to-date book this volume is the only one to cover the de facto standard implementation from the 4.4BSD-Lite release, the foundation for TCP/IP implementations run daily on hundreds of thousands of systems worldwide. Combining 500 illustrations with 15,000 lines of real, working code, *TCP/IP Illustrated, Volume 2* uses a teach-by-example approach to help you master TCP/IP implementation. You will learn about such topics as the relationship between the sockets API and the protocol suite, and the differences between a host implementation and a router. In addition, the book covers the newest features of the 4.4BSD-Lite release, including multicasting, long fat pipe support, window scale, timestamp options, and protection against wrapped sequence numbers, and many other topics. Comprehensive in scope, based on a working standard, and thoroughly illustrated, this book is an indispensable resource for anyone working with TCP/IP.

The Art of UNIX Programming Pearson Education

This book addresses the recent developments in systems maintenance research and practices ranging from technicality of systems evolution to managerial aspects of the topic, including issues such as evolving legacy systems to e-business, applying patterns for reengineering legacy systems to web, architectural recovery of legacy systems, evolving legacy systems into software components.

Tcl/Tk 8.5 Programming Cookbook Pearson Education

"Every C++ professional needs a copy of *Effective C++*. It is an absolute must-read for anyone thinking of doing serious C++ development. If you've never read *Effective C++* and you think you know everything about C++, think again." — Steve Schirripa, Software Engineer, Google "C++ and the C++ community have grown up in the last fifteen years, and the third edition of *Effective C++* reflects this. The clear and precise style of the book is evidence of Scott's deep insight and distinctive ability to impart knowledge." — Gerhard Kreuzer, Research and Development Engineer, Siemens AG The first two editions of *Effective C++* were embraced by hundreds of thousands of programmers worldwide. The reason is clear: Scott Meyers' practical approach to C++ describes the rules of thumb used by the experts — the things they almost always do or almost always avoid doing — to produce clear, correct, efficient code. The book is organized around 55 specific guidelines, each of which describes a way to write better C++. Each is backed by concrete examples. For this third edition, more than half the content is new, including added chapters on managing resources and using templates. Topics from the second edition have been extensively revised to reflect modern design considerations, including exceptions, design patterns, and multithreading. Important features of *Effective C++* include: Expert guidance on the design of effective classes, functions, templates, and inheritance hierarchies. Applications of new "TR1"

standard library functionality, along with comparisons to existing standard library components. Insights into differences between C++ and other languages (e.g., Java, C#, C) that help developers from those languages assimilate "the C++ way" of doing things.

Programming in Lua Springer Science & Business Media

Machine generated contents note: Chapter 1: Tcl/Tk Features Chapter 2: The Mechanics of Using the Tcl and Tk Interpreters Chapter 3: Introduction to the Tcl Language Chapter 4: File System, Disk I/O and Sockets Chapter 5: Using Strings and Lists Chapter 6: Basic list, array and dict Chapter 7: Advanced List, array and dict Chapter 8: Procedure Techniques Chapter 9: Namespaces Chapter 10: Basic TclOO Chapter 11: Advanced TclOO Chapter 12: Packages and modules Chapter 13: Introduction to Tk Graphics Chapter 14: Overview of the canvas Widget Chapter 15: The text widget and htmllib Chapter 16: Themed Widgets Chapter 17: Tk Megawidgets Chapter 18: Writing a Tcl Extension Chapter 19: Extensions and Packages Chapter 20: Programming Tools Chapter 21: Debugging and Optimization techniques Chapter 22: Tips and Techniques .

Learn to create modern GUIs using Tkinter by building real-world projects in Python CYBOP

Python 3 is the best version of the language yet: It is more powerful, convenient, consistent, and expressive than ever before. Now, leading Python programmer Mark Summerfield demonstrates how to write code that takes full advantage of Python 3's features and idioms. The first book written from a completely "Python 3" viewpoint, *Programming in Python 3* brings together all the knowledge you need to write any program, use any standard or third-party Python 3 library, and create new library modules of your own. Summerfield draws on his many years of Python experience to share deep insights into Python 3 development you won't find anywhere else. He begins by illuminating Python's "beautiful heart": the eight key elements of Python you need to write robust, high-performance programs. Building on these core elements, he introduces new topics designed to strengthen your practical expertise—one concept and hands-on example at a time. This book's coverage includes Developing in Python using procedural, object-oriented, and functional programming paradigms Creating custom packages and modules Writing and reading binary, text, and XML files, including optional compression, random access, and text and XML parsing Leveraging advanced data types, collections, control structures, and functions Spreading program workloads across multiple processes and threads Programming SQL databases and key-value DBM files Utilizing Python's regular expression mini-language and module Building usable, efficient, GUI-based applications Advanced programming techniques, including generators, function and class decorators, context managers, descriptors, abstract base classes, metaclasses, and more *Programming in Python 3* serves as both tutorial and language reference, and it is accompanied by extensive downloadable example code—all of it tested with the final version of Python 3 on Windows, Linux, and Mac OS X.

Writing Better Programs with Tcl and Tk Addison Wesley Longman

Effective Tcl/Tk Programming Writing Better Programs with Tcl and Tk Addison-Wesley Professional

Effective C++ *Effective Tcl/Tk Programming* Writing Better Programs with Tcl and Tk In just a few chapters you will learn about Tcl features that allow you to isolate and protect your code from being damaged in large applications. You will even learn how to extend the language itself. *Tcl/Tk: A Developer's Guide* clearly discusses development tools, proven techniques, and

existing extensions. It shows how to use Tcl/Tk effectively and provides many code examples. This fully revised new edition is the complete resource for computer professionals, from systems administrators to programmers. It covers versions 7.4 to 8.4 and includes a CD-ROM containing the interpreters, libraries, and tutorials to get you started quickly. Additional materials in the book include case studies and discussions of techniques for the advanced user. On the CD-ROM

- *Distributions for Tcl 8.3 and 8.4 for Linux, Solaris, Macintosh, and Windows.
- *A copy of ActiveTcl from ActiveState.
- *The latest release of TclTutor.
- *How-to's and tutorials as well as copies of all the tools discussed in the book.

[An Investigation on the Applicability of Inter-disciplinary Concepts to Software System Development](#)
Packt Publishing Ltd

Introduction to Network Simulator NS2 is a primer providing materials for NS2 beginners, whether students, professors, or researchers for understanding the architecture of Network Simulator 2 (NS2)

Related with Effective Tcl Tk Programming Writing Better Programs In Tcl And Tk:

- Woman Within Size Guide : [click here](#)

and for incorporating simulation modules into NS2. The authors discuss the simulation architecture and the key components of NS2 including simulation-related objects, network objects, packet-related objects, and helper objects. The NS2 modules included within are nodes, links, SimpleLink objects, packets, agents, and applications. Further, the book covers three helper modules: timers, random number generators, and error models. Also included are chapters on summary of debugging, variable and packet tracing, result compilation, and examples for extending NS2. Two appendices provide the details of scripting language Tcl, OTcl and AWK, as well object oriented programming used extensively in NS2.

[Programming in Python 3](#) Frontiers Media SA

Here is a programmer's guide to using and programming POSIX threads, commonly known as Pthreads. A "coder's book", this title tells how to use Pthreads in the real world, making efficient and portable applications. Pthreads are an important set of current tools programmers need to have in today's network-intensive climate.