

Sams Teach Yourself Cobol In 24 Hours

Sams Teach Yourself COBOL in 24 Hours: A Comprehensive Review

Learning to code can feel daunting, but the promise of rapid acquisition is alluring. This is precisely the appeal of "Sams Teach Yourself COBOL in 24 Hours," a book designed for beginners seeking a quick entry into the world of COBOL programming. This review will explore the book's features, its effectiveness as a learning tool, and its relevance in today's technological landscape, touching upon keywords like **COBOL programming tutorials**, **COBOL for beginners**, **quick COBOL learning**, **legacy system maintenance**, and **COBOL programming language**.

Introduction to COBOL and the "24-Hour" Approach

COBOL, or Common Business-Oriented Language, is a remarkably persistent programming language. Despite its age, COBOL remains crucial for many large-scale business applications and legacy systems. Many organizations still rely on COBOL systems for financial transactions, inventory management, and other critical functions. This makes understanding COBOL, even for a quick overview, a valuable skill. "Sams Teach Yourself COBOL in 24 Hours" aims to provide that overview, promising a fast track to basic COBOL proficiency. The "24-hour" claim is, of course, a marketing simplification; mastery of any programming language takes significantly longer. However, the book's

structure is designed for rapid learning, focusing on core concepts and practical application.

What the Book Offers: Content and Approach

The book follows a structured, tutorial-based approach. Each "hour" represents a chapter focusing on a specific aspect of COBOL. This includes covering fundamental elements such as data types, variables, arithmetic operations, and input/output procedures. It progresses to more advanced topics, depending on the specific edition, such as file handling, structured programming techniques, and potentially even object-oriented COBOL extensions. The "Sams Teach Yourself" series generally emphasizes a hands-on approach, encouraging readers to write and execute code from the start. This practical focus, supported by numerous examples and exercises, is a significant strength of the book. It directly addresses the needs of those seeking a **COBOL programming tutorial** designed for swift learning.

Benefits and Limitations of the "24-Hour" Method

The "Sams Teach Yourself COBOL in 24 Hours" approach, while efficient in presenting foundational knowledge, has inherent limitations. Twenty-four hours are insufficient for developing true mastery of COBOL or any programming language. The book prioritizes breadth over depth; it introduces many concepts but might not allow sufficient time for in-depth understanding and practical application of each. For this reason, the book serves best as an introduction, a stepping stone to more advanced learning resources, and perhaps as a suitable option for those who need only a superficial understanding of COBOL for specific tasks. Therefore, those seeking **quick COBOL learning** will find the book's concise structure beneficial.

Practical Applications and Relevance of COBOL Today

Despite the rise of modern languages, COBOL remains relevant in maintaining and updating legacy systems. Many businesses still rely on COBOL-based applications, creating a demand for skilled COBOL programmers. The need for **legacy system maintenance** is a significant factor driving the continued interest in COBOL. Understanding COBOL can open doors to job opportunities in IT maintenance and modernization projects. For instance, a programmer might be tasked with modifying a decades-old COBOL program to integrate with a modern database or to improve its efficiency. Therefore, the book's focus on fundamental COBOL skills can be a valuable asset in this niche.

Conclusion: A Useful Introduction, Not a Panacea

"Sams Teach Yourself COBOL in 24 Hours" provides a streamlined introduction to COBOL programming. It's not a comprehensive guide, and the "24-hour" timeframe is a significant simplification. However, its structured approach, practical examples, and focus on hands-on learning make it an effective entry point for beginners. It provides a solid foundation in the fundamentals of COBOL and can be a valuable resource for those seeking a swift overview of the language. While not a replacement for thorough, in-depth study, it serves as an excellent starting point for those looking into **COBOL programming language** learning. It's particularly helpful for those involved with **COBOL for beginners** seeking a rapid introduction to this sometimes-overlooked language.

FAQ

Q1: Is "Sams Teach Yourself COBOL in 24 Hours" suitable for complete beginners?

A1: Yes, the book is specifically designed for beginners with little to no programming experience. It starts with the

absolute basics and gradually introduces more complex concepts. However, prior exposure to programming logic would be beneficial.

Q2: How much COBOL can I realistically learn in 24 hours?

A2: The "24 hours" is a marketing term; you won't become a COBOL expert in a day. You can, however, expect to grasp fundamental concepts, understand basic syntax, and be able to write simple COBOL programs.

Q3: What are the system requirements to use the book effectively?

A3: You'll need a COBOL compiler and an Integrated Development Environment (IDE). The specific requirements depend on the edition of the book and the compiler used. The book will likely provide guidance on setting up your development environment.

Q4: Does the book cover object-oriented COBOL?

A4: This depends on the edition of the book. Older editions focus primarily on procedural COBOL. More recent editions might include an introduction to object-oriented COBOL extensions, but the coverage may be limited due to the book's brevity.

Q5: Are there any online resources that complement the book?

A5: Many online resources are available, including tutorials, documentation, and online forums dedicated to COBOL programming. These can enhance your learning experience alongside the book.

Q6: Is this book enough to get a job as a COBOL programmer?

A6: No, this book is just an introductory guide. To become a working COBOL programmer, you will need to expand your knowledge substantially beyond what this book covers. Extensive practice and experience with more complex programs are necessary.

Q7: How does the book compare to other COBOL learning resources?

A7: Compared to more comprehensive COBOL textbooks, it sacrifices depth for brevity. It's ideal for a quick overview or as a starting point before diving into more detailed resources.

Q8: What are the current job prospects for COBOL programmers?

A8: While not as abundant as for other programming languages, there's still a significant demand for experienced COBOL programmers, especially for maintenance and modernization of legacy systems. The demand is expected to remain for years to come due to the vast number of existing COBOL applications.

Can You Really Learn COBOL in 24 Hours? A Deep Dive into "Sams Teach Yourself COBOL in 24 Hours"

The alluring promise of mastering a complex programming language like COBOL legacy in just 24 hours is undeniably attractive. Books like "Sams Teach Yourself COBOL in 24 Hours" capitalize on this longing for rapid skill acquisition, but the reality is far more nuanced. This article will delve into the statements of such accelerated learning guides, examining their advantages and deficiencies to provide a balanced perspective on what you can realistically foresee from this type of guidance.

The "Sams Teach Yourself" series has a long tradition of offering concise, practical guides for a wide range of topics.

While the 24-hour structure implies a breakneck learning journey, it's crucial to understand what this signifies. The book doesn't promise proficiency, but rather a foundational understanding of COBOL's form. Think of it as a express overview, a preview to whet your appetite, not a comprehensive course.

The book's technique typically involves a organized breakdown of COBOL principles, often using short, succinct explanations and numerous instances. Each "hour" deals with a specific topic, developing upon previous information. Early chapters present fundamental elements like data sorts, variables, and basic commands. Later chapters explore more advanced features such as file handling, functions, and potentially some basic database interactions.

However, the tempo of this approach inevitably means that many subtleties are glossed over. The 24-hour framework severely limits the opportunity for drill, consolidation, and in-depth exploration of advanced concepts. This constrains its value as a standalone learning resource for developing genuine COBOL programming abilities.

Furthermore, COBOL's legacy nature presents additional obstacles. The language is considerably different from more modern idioms like Python or Java, requiring a shift in outlook. Its verbosity also contributes to a steeper learning curve than many expect.

Therefore, "Sams Teach Yourself COBOL in 24 Hours" serves best as an beginning to the language, a starting point for further exploration. It's ideal for those who need a fast overview, perhaps to understand legacy systems or to gauge their interest in the language before investing in more intensive learning. Think of it as a sampler - it offers a taste of what COBOL entails but does not provide the skills necessary for independent development of complex applications.

To effectively learn COBOL, supplementing this quick learning guide with more in-depth resources is essential. This could involve online lessons, further research, and, most importantly, exercise. Building small systems and actively working with COBOL code is the most effective way to consolidate your understanding and acquire practical

competencies.

In conclusion, "Sams Teach Yourself COBOL in 24 Hours" isn't a instant bullet for COBOL expertise. It offers a concise initiation, perfect for getting a feel for the language or gaining a elementary comprehension. However, serious programmers will need to allocate significantly more time and effort to achieve genuine proficiency. Supplementing the book with further learning and functional experience is paramount.

Frequently Asked Questions (FAQs):

1. **Q: Is "Sams Teach Yourself COBOL in 24 Hours" enough to get a COBOL job?** A: No. This book provides a basic introduction. A job requires significantly more extensive knowledge and practical experience.
2. **Q: Can I learn COBOL solely from this book?** A: While you can gain a basic understanding, it is not sufficient for proficient programming. Supplementing with additional resources is crucial.
3. **Q: Is COBOL still relevant in today's tech landscape?** A: Yes, surprisingly. Many legacy systems still rely on COBOL, creating a demand for skilled professionals in maintenance and modernization.
4. **Q: What is the best way to learn COBOL after finishing this book?** A: Seek out online courses, practice coding regularly, and consider contributing to open-source projects to gain experience.

https://xs.full doc.me/fp172609/8759F844P9140050/TEXT/home__exercise-guide.pdf

https://xs.full doc.me/140050/D71469R966/PAGE/making-nations__creating__strangers-african-social__studies__series__pdf

https://xs.full doc.me/140050/43VU798/PPT/epson-aculaser_c9100__service__manual-repair__guide.pdf

https://xs.full doc.me/ht/99T635H059260917/EPDF/range__management_principles__and_practices__6th-edition.pdf

https://xs.fullloc.me/140050/25183W213N/ITUNE/quality_manual__example.pdf

https://xs.fullloc.me/mz/723ZM62738260917/EPDF/strategic__management_concepts_frank__rothaermel.pdf

https://xs.fullloc.me/50140VK/140050170926/CHAPTER/campbell_biology-8th-edition__test_bank-free.pdf

https://xs.fullloc.me/X77846761I/X85686I/EBOOK/frick_screw__compressor_service_manual.pdf

https://xs.fullloc.me/iv/56441IV/ITUNE/new-holland_tg210-tg230-tg255_tg285-tractors_service-workshop__manual__download.pdf

https://xs.fullloc.me/vj/90V8803J37260917/RTF/manual-philips_matchline__tv.pdf