cs vs software engineering reddit

cs vs software engineering reddit is a popular topic of discussion among students, professionals, and enthusiasts looking to understand the distinctions and overlaps between computer science and software engineering. On Reddit, numerous threads explore career prospects, curriculum differences, job roles, and industry expectations for both fields. This article delves into the key insights shared by the Reddit community, highlighting the pros and cons of each discipline, and providing a comprehensive comparison to help readers make informed decisions. By analyzing various perspectives, this piece covers educational pathways, skill requirements, and potential career trajectories associated with computer science and software engineering. The discussion also touches on how Reddit users perceive the practical applications, theoretical foundations, and industry relevance of both domains. Below is a structured overview of the main points covered in this detailed comparison.

- Educational Differences Between Computer Science and Software Engineering
- Career Opportunities and Industry Demand
- Skill Sets and Technical Competencies
- Salary Expectations and Job Satisfaction
- Common Misconceptions and Community Opinions on Reddit

Educational Differences Between Computer Science and Software Engineering

Understanding the educational distinctions is crucial for anyone weighing the cs vs software engineering reddit debate. Computer science (CS) programs typically emphasize theoretical foundations, including algorithms, data structures, computation theory, and mathematics. In contrast, software engineering (SE) curricula focus more on the practical aspects of software development, project management, and engineering principles.

Core Curriculum Focus

Reddit users often highlight that computer science degrees prioritize understanding the underlying principles of computing, which includes courses on operating systems, compiler design, and artificial intelligence. Software engineering programs tend to incorporate classes on software lifecycle models, quality assurance, testing methodologies, and collaborative development practices.

Hands-On Experience

According to Reddit discussions, software engineering students generally engage in more team-based projects and real-world application development as part of their coursework. This practical orientation prepares them for the collaborative nature of software development in industry settings. Computer science students may have fewer mandatory group projects but often participate in research or independent programming challenges that deepen their problem-solving skills.

Career Opportunities and Industry Demand

The career trajectories of computer science and software engineering graduates often overlap, yet distinct paths exist. The Reddit community frequently discusses how the job market perceives these degrees and the types of roles each prepares candidates for.

Typical Job Roles for Computer Science Graduates

Graduates with a computer science background may pursue roles such as software developer, data scientist, systems analyst, or research scientist. Theoretical knowledge gained during a CS degree enables opportunities in fields like machine learning, cybersecurity, and academia.

Common Roles for Software Engineering Graduates

Software engineering graduates are commonly employed as software engineers, quality assurance engineers, DevOps engineers, and project managers. Their expertise in software development processes and engineering standards positions them well for roles that emphasize building reliable, maintainable software systems.

Industry Demand Trends

Reddit discussions reveal that both disciplines enjoy strong demand, but software engineering roles often have more immediate openings due to the emphasis on product development and delivery. Computer science graduates may find niche opportunities in emerging technologies and research-heavy positions.

Skill Sets and Technical Competencies

Analyzing the skill sets associated with cs vs software engineering reddit reveals differences in technical proficiencies and mindset. Both fields require programming skills, but the depth and focus vary.

Programming and Technical Skills

Computer science students typically gain proficiency in multiple programming languages and develop strong analytical abilities to solve complex problems. They often learn low-level programming, algorithms optimization, and computational theory. Software engineering students focus on coding best practices, software design patterns, version control systems, and continuous integration tools.

Soft Skills and Collaboration

Software engineering programs emphasize teamwork, communication, and project management skills, which are essential in collaborative software development environments. Reddit users note that these soft skills can differentiate software engineers in the workplace and improve project outcomes.

Problem-Solving Approaches

While both fields require problem-solving skills, computer science leans towards abstract and theoretical problem-solving, whereas software engineering applies problem-solving to practical software design and implementation challenges.

Salary Expectations and Job Satisfaction

Salary and job satisfaction are commonly debated topics in cs vs software engineering reddit threads. Understanding compensation trends and workplace satisfaction can help prospective students and professionals make career decisions.

Salary Comparisons

Reddit users often report that salaries for computer science and software engineering roles are comparable, with slight variations depending on location, experience, and specialization. Software engineers working in large tech companies or startups may see competitive pay packages, while computer scientists in research or specialized roles might have varied salary ranges.

Job Satisfaction Factors

Job satisfaction depends more on the work environment, company culture, and personal interest than the degree itself. Reddit discussions emphasize that passion for the work, opportunities for growth, and the nature of projects significantly influence satisfaction levels for both computer scientists and software engineers.

Work-Life Balance

Many Reddit contributors highlight that work-life balance can vary widely within both fields. Software engineers may face tight deadlines and high-pressure environments, while computer scientists in research or academia might experience different stressors. Ultimately, individual roles and employers play a significant role.

Common Misconceptions and Community Opinions on Reddit

The Reddit community has identified several misconceptions concerning cs vs software engineering that often appear in discussions, clarifying the realities of both fields.

Misconception: Computer Science is Only About Programming

Many believe computer science solely involves coding, but Reddit users clarify that CS encompasses a broad range of topics, including theory, hardware, and abstract computation. Programming is a tool rather than the entire focus.

Misconception: Software Engineering is Less Theoretical

While software engineering is more applied, it still requires understanding software architecture, design principles, and engineering rigor. Reddit threads emphasize that SE is a disciplined engineering field, not just coding.

Community Advice for Prospective Students

Redditors suggest prospective students consider their interests and career goals carefully. Those fascinated by theory and research might prefer computer science, while individuals eager to build and deliver software products may gravitate toward software engineering. Exploration of course offerings and internships is recommended to gain clarity.

Summary of Reddit Consensus

- Both fields offer valuable and overlapping career options.
- Educational focus varies between theory (CS) and application (SE).
- Soft skills and teamwork are more emphasized in software engineering programs.
- Salary differences are minimal and depend on role and experience.
- Choosing between CS and SE depends on personal interests, not just job prospects.

Frequently Asked Questions

What are the main differences between Computer Science and Software Engineering as discussed on Reddit?

On Reddit, users often highlight that Computer Science focuses more on theoretical foundations, algorithms, and computation, while Software Engineering emphasizes practical application, software development processes, and project management.

Which degree is considered more versatile for tech careers according to Reddit discussions: CS or Software Engineering?

Reddit users generally agree that a Computer Science degree is more versatile due to its strong theoretical foundation, allowing graduates to work in diverse fields like AI, data science, and research, whereas Software Engineering is more specialized towards software development roles.

Do Reddit users believe Software Engineering is easier than Computer Science?

Many Redditors suggest that Software Engineering can be perceived as more practical and applied, sometimes considered easier by those who prefer hands-on coding and projects, whereas Computer Science involves more challenging math and theory, but this varies by individual preference and program.

How do career opportunities compare between CS and Software Engineering according to Reddit threads?

Reddit discussions often indicate that both degrees lead to good career opportunities in tech, but CS graduates might have broader options including research and specialized tech roles, while Software Engineering graduates are typically well-prepared for software development and engineering positions in industry.

Is it possible to switch from a Computer Science background to Software Engineering roles based on Reddit advice?

Yes, Reddit users frequently mention that transitioning from a CS background to software engineering roles is common and feasible by gaining practical coding experience, understanding software development lifecycle, and working on projects or internships to build relevant skills.

Additional Resources

1. *"Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin*
This book is a classic in the software engineering community, emphasizing the importance of writing

readable, maintainable, and efficient code. It provides practical advice and best practices for software developers to improve their coding skills. The author uses real-world examples to demonstrate how to transform messy code into clean, elegant solutions. It's highly recommended for both computer science students and professional software engineers.

2. *“The Pragmatic Programmer: Your Journey to Mastery” by Andrew Hunt and David Thomas*
A foundational text for software engineers, this book covers a wide range of topics from personal responsibility to architectural techniques. It bridges the gap between computer science theory and practical software development. Through anecdotes and tips, it encourages a pragmatic approach to coding and engineering challenges, making it relevant to discussions on Reddit about CS versus software engineering.

3. *“Introduction to the Theory of Computation” by Michael Sipser*
This book is often recommended for computer science students interested in the theoretical underpinnings of computation. It covers essential topics like automata theory, complexity, and computability. While more academic than practical, it provides a strong foundation that complements software engineering knowledge, often discussed in debates on Reddit about the differences between CS and SE.

4. *“Software Engineering: A Practitioner's Approach” by Roger S. Pressman and Bruce R. Maxim*
This comprehensive textbook is widely used in software engineering courses and covers the software development lifecycle, process models, and project management. It offers in-depth insights into software design, testing, and maintenance. The book is ideal for those looking to understand the structured methodologies behind professional software engineering practices.

5. *“Code Complete: A Practical Handbook of Software Construction” by Steve McConnell*
Focusing on software construction, this book provides detailed techniques and strategies for writing high-quality code. It synthesizes best practices from industry and academic research to improve software development. Readers from both CS and SE backgrounds find it valuable for bridging theory and practice.

6. *“Design Patterns: Elements of Reusable Object-Oriented Software” by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides*
Known as the “Gang of Four” book, it introduces fundamental design patterns used in software engineering. These patterns help solve common design problems and improve code maintainability and scalability. It is a staple for software engineers and computer scientists alike, often referenced in technical discussions on Reddit.

7. *“Algorithms” by Robert Sedgewick and Kevin Wayne*
A detailed and accessible guide to algorithms, this book covers essential data structures and algorithmic techniques. It balances theoretical concepts with practical implementation, making it useful for both CS students and software engineers. The book’s clarity and depth make it a frequent recommendation in online discussions comparing CS theory and software engineering practice.

8. *“Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation” by Jez Humble and David Farley*
This book focuses on modern software engineering practices related to continuous integration and continuous delivery. It explains how to automate the software release process to improve quality and speed. It is particularly relevant for software engineers interested in DevOps and agile methodologies, topics often debated on Reddit.

9. *“Structure and Interpretation of Computer Programs” by Harold Abelson and Gerald Jay Sussman* A seminal text in computer science education, this book introduces fundamental programming concepts through Scheme. It emphasizes abstraction, recursion, and interpreters, building a deep understanding of programming languages. Though more theoretical, it is highly regarded by both CS students and software engineers who want to deepen their conceptual knowledge.

Cs Vs Software Engineering Reddit

Find other PDF articles:
https://test.murphyjewelers.com/archive-library-503/pdf?docid=bMs42-2992&title=maysville-ky-health-department.pdf

cs vs software engineering reddit: Mining Software Engineering Data for Software Reuse Themistoklis Diamantopoulos, Andreas L. Symeonidis, 2020-03-30 This monograph discusses software reuse and how it can be applied at different stages of the software development process, on different types of data and at different levels of granularity. Several challenging hypotheses are analyzed and confronted using novel data-driven methodologies, in order to solve problems in requirements elicitation and specification extraction, software design and implementation, as well as software quality assurance. The book is accompanied by a number of tools, libraries and working prototypes in order to practically illustrate how the phases of the software engineering life cycle can benefit from unlocking the potential of data. Software engineering researchers, experts, and practitioners can benefit from the various methodologies presented and can better understand how knowledge extracted from software data residing in various repositories can be combined and used to enable effective decision making and save considerable time and effort through software reuse. Mining Software Engineering Data for Software Reuse can also prove handy for graduate-level students in software engineering.

cs vs software engineering reddit: Computing and Technology Ethics Emanuelle Burton, Judy Goldsmith, Nicholas Mattei, Cory Siler, Sara-Jo Swiatek, 2023-02-14 A new approach to teaching computing and technology ethics using science fiction stories. Should autonomous weapons be legal? Will we be cared for by robots in our old age? Does the efficiency of online banking outweigh the risk of theft? From communication to travel to medical care, computing technologies have transformed our daily lives, for better and for worse. But how do we know when a new development comes at too high a cost? Using science fiction stories as case studies of ethical ambiguity, this engaging textbook offers a comprehensive introduction to ethical theory and its application to contemporary developments in technology and computer science. Computing and Technology Ethics: Engaging through Science Fiction first introduces the major ethical frameworks: deontology, utilitarianism, virtue ethics, communitarianism, and the modern responses of responsibility ethics, feminist ethics, and capability ethics. It then applies these frameworks to many of the modern issues arising in technology ethics including privacy, computing, and artificial intelligence. A corresponding anthology of science fiction brings these quandaries to life and challenges students to ask ethical questions of themselves and their work. Uses science fiction case studies to make ethics education engaging and fun Trains students to recognize, evaluate, and respond to ethical problems as they arise Features anthology of short stories from internationally acclaimed writers including Ken Liu, Elizabeth Bear, Paolo Bacigalupi, and T. C. Boyle to animate ethical challenges in computing technology Written by interdisciplinary author team of computer scientists and ethical theorists Includes a robust suite of instructor resources, such as pedagogy guides, story frames, and reflection questions

cs vs software engineering reddit: Software Engineering: Emerging Trends and Practices in System Development Radek Silhavy, Petr Silhavy, 2025-08-11 This book discovers peer-reviewed research from an international research conference that unites experts in software engineering, data science, artificial intelligence, cybernetics, and informatics. This book presents cutting-edge methods, practical case studies, and foundational advances that address real-world challenges across the computational spectrum. Whether you seek rigorous theory, proven development practices, or visionary perspectives on emerging technologies, this book provides a comprehensive resource for researchers, practitioners, and students committed to shaping the future of digital systems.

cs vs software engineering reddit: Software Engineering Perspectives in Systems Radek Silhavy, 2022-07-16 The study of software engineering and its applications to system engineering is critical in computer science research. Modern research methodologies, as well as the use of machine and statistical learning in software engineering research, are covered in this book. This book contains the refereed proceedings of the Software Engineering Perspectives in Systems part of the 11th Computer Science On-line Conference 2022 (CSOC 2022), which was held in April 2022 online.

cs vs software engineering reddit: *Computational Methods in Science and Technology* Sukhpreet Kaur, Sushil Kamboj, Manish Kumar, Arvind Dagur, Dhirendra Kumar Shukla, 2024-10-10 This book contains the proceedings of the 4TH International Conference on Computational Methods in Science and Technology (ICCMST 2024). The proceedings explores research and innovation in the field of Internet of things, Cloud Computing, Machine Learning, Networks, System Design and Methodologies, Big Data Analytics and Applications, ICT for Sustainable Environment, Artificial Intelligence and it provides real time assistance and security for advanced stage learners, researchers and academicians has been presented. This will be a valuable read to researchers, academicians, undergraduate students, postgraduate students, and professionals within the fields of Computer Science, Sustainability and Artificial Intelligence.

cs vs software engineering reddit: Computer Science Engineering and Emerging Technologies Rajeev Sobti, Rachit Garg, Ajeet Kumar Srivastava, Gurpeet Singh Shahi, 2024-06-07 The year 2022 marks the 100th birth anniversary of Kathleen Hylda Valerie Booth, who wrote the first assembly language and designed the assembler and auto code for the first computer systems at Birkbeck College, University of London. She helped design three different machines including the ARC (Automatic Relay Calculator), SEC (Simple Electronic Computer), and APE(X). School of Computer Science and Engineering, under the aegis of Lovely Professional University, pays homage to this great programmer of all times by hosting “BOOTH100”—6th International Conference on Computing Sciences.

cs vs software engineering reddit: Python Natural Language Processing Cookbook Zhenya Antić, Saurabh Chakravarty, 2024-09-13 Updated to include three new chapters on transformers, natural language understanding (NLU) with explainable AI, and dabbling with popular LLMs from Hugging Face and OpenAI Key Features Leverage ready-to-use recipes with the latest LLMs, including Mistral, Llama, and OpenAI models Use LLM-powered agents for custom tasks and real-world interactions Gain practical, in-depth knowledge of transformers and their role in implementing various NLP tasks with open-source and advanced LLMs Purchase of the print or Kindle book includes a free PDF eBook Book DescriptionHarness the power of Natural Language Processing (NLP) to overcome real-world text analysis challenges with this recipe-based roadmap written by two seasoned NLP experts with vast experience transforming various industries with their NLP prowess. You’ll be able to make the most of the latest NLP advancements, including large language models (LLMs), and leverage their capabilities through Hugging Face transformers. Through a series of hands-on recipes, you’ll master essential techniques such as extracting entities and visualizing text data. The authors will expertly guide you through building pipelines for sentiment analysis, topic modeling, and question-answering using popular libraries like spaCy, Gensim, and NLTK. You’ll also learn to implement RAG pipelines to draw out precise answers from a text corpus using LLMs. This second edition expands your skillset with new chapters on cutting-edge LLMs like GPT-4, Natural Language Understanding (NLU), and Explainable AI (XAI)—fostering trust in your NLP models. By the end of this book, you'll be equipped with the skills to apply advanced text processing techniques, use pre-trained transformer models, build custom NLP pipelines to extract valuable insights from text data to drive informed decision-making.What you will learn Understand fundamental NLP concepts along with their applications using examples in Python Classify text quickly and accurately with rule-based and supervised methods Train NER models and perform sentiment analysis to identify entities and emotions in text Explore topic modeling and text visualization to reveal themes and relationships within text Leverage Hugging Face and OpenAI LLMs to perform advanced NLP tasks Use question-answering techniques to handle both open and closed domains Apply XAI techniques to better understand your model predictions Who this book is for This updated edition of the Python Natural Language Processing Cookbook is for data scientists, machine learning engineers, and developers with a background in Python. Whether you’re looking to learn NLP techniques, extract valuable insights from textual data, or create foundational applications, this book will equip you with basic to intermediate skills. No prior NLP knowledge is necessary to get started. All you need is familiarity with basic programming principles. For seasoned developers, the updated sections offer the latest on transformers, explainable AI, and Generative AI with LLMs.

cs vs software engineering reddit: ,

cs vs software engineering reddit: Artificial Intelligence Methods for Optimization of the Software Testing Process Sahar Tahvili, Leo Hatvani, 2022-07-21 Artificial Intelligence Methods for Optimization of the Software Testing Process: With Practical Examples and Exercises presents different AI-based solutions for overcoming the uncertainty found in many initial testing problems. The concept of intelligent decision making is presented as a multi-criteria, multi-objective undertaking. The book provides guidelines on how to manage diverse types of uncertainty with intelligent decision-making that can help subject matter experts in many industries improve various processes in a more efficient way. As the number of required test cases for testing a product can be large (in industry more than 10,000 test cases are usually created). Executing all these test cases without any particular order can impact the results of the test execution, hence this book fills the need for a comprehensive resource on the topics on the how's, what's and whys. To learn more about Elsevier's Series, Uncertainty, Computational Techniques and Decision Intelligence, please visit this link: https://www.elsevier.com/books-and-journals/book-series/uncertainty-computational-techniques-and-decision-intelligence - Presents one of the first empirical studies in the field, contrasting theoretical assumptions on innovations in a real industrial environment with a large set of use cases from developed and developing testing processes at various large industries - Explores specific comparative methodologies, focusing on developed and developing AI-based solutions - Serves as a guideline for conducting industrial research in the artificial intelligence and software testing domain - Explains all proposed solutions through real industrial case studies

cs vs software engineering reddit: *Supercharged Coding with GenAI* Hila Paz Herszfang, Peter V. Henstock, 2025-08-28 Unlock the power of generative AI in Python development and learn how you can enhance your coding speed, quality, and efficiency with real-world examples and hands-on strategies Key Features Discover how GitHub Copilot, ChatGPT, and the OpenAI API can boost your coding productivity Push beyond the basics to apply advanced techniques across the software development lifecycle Master best practices and advanced techniques to achieve quality code for even complex tasks Purchase of the print or Kindle book includes a free PDF eBook Book DescriptionSoftware development is being transformed by GenAI tools, such as ChatGPT, OpenAI API, and GitHub Copilot, redefining how developers work. This book will help you become a power user of GenAI for Python code generation, enabling you to write better software faster. Written by an ML advisor with a thriving tech social media presence and a top AI leader who brings Harvard-level instruction to the table, this book combines practical industry insights with academic expertise. With this book, you'll gain a deep understanding of large language models (LLMs) and develop a systematic approach to solving complex tasks with AI. Through real-world examples and practical exercises, you’ll master best practices for leveraging GenAI, including prompt engineering techniques like few-shot learning and Chain-of-Thought (CoT). Going beyond simple code generation, this book teaches you how to automate debugging, refactoring, performance optimization, testing, and monitoring. By applying reusable prompt frameworks and AI-driven workflows, you’ll streamline your software development lifecycle (SDLC) and produce high-quality, well-structured code. By the end of this book, you'll know how to select the right AI tool for each task, boost efficiency, and anticipate your next coding moves—helping you stay ahead in the AI-powered development era.What you will learn Work with GitHub Copilot in PyCharm, VS Code, and Jupyter Notebook Apply advanced prompting methods with ChatGPT and OpenAI API Gain insight into GenAI fundamentals to achieve better outcomes Adopt our structured framework to produce high-quality code Find out how to select the optimal GenAI tool for solving your specific tasks Elevate your use of GenAI tools from debugging to delivery Join the next generation of supercharged software engineers Who this book is for If you are a Python developer curious about GenAI and are looking to elevate your software engineering productivity, Supercharged Coding with GenAI will transform your approach to software. Covering various structured examples of varying problem complexities that showcase the use of advanced prompting techniques, this book is suitable for early intermediate through advanced developers. To get the most out of this book, you should have at least one year of hands-on Python development experience and be somewhat familiar with the SDLC.

cs vs software engineering reddit: *Computational Science and Its Applications – ICCSA 2022 Workshops* Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, Chiara Garau, 2022-07-25 The eight-volume set LNCS 13375 – 13382 constitutes the proceedings of the 22nd International Conference on Computational Science and Its Applications, ICCSA 2022, which was held in Malaga, Spain during July 4 – 7, 2022. The first two volumes contain the proceedings from ICCSA 2022, which are the 57 full and 24 short papers presented in these books were carefully reviewed and selected from 279 submissions. The other six volumes present the workshop proceedings, containing 285 papers out of 815 submissions. These six volumes includes the proceedings of the following workshops: Advances in Artificial Intelligence Learning Technologies: Blended Learning, STEM, Computational Thinking and Coding (AAILT 2022); Workshop on Advancements in Applied Machine-learning and Data Analytics (AAMDA 2022); Advances in information Systems and Technologies for Emergency management, risk assessment and mitigation based on the Resilience (ASTER 2022); Advances in Web Based Learning (AWBL 2022); Blockchain and Distributed Ledgers: Technologies and Applications (BDLTA 2022); Bio and Neuro inspired Computing and Applications (BIONCA 2022); Configurational Analysis For Cities (CA Cities 2022); Computational and Applied Mathematics (CAM 2022), Computational and Applied Statistics (CAS 2022); Computational Mathematics, Statistics and Information Management (CMSIM); Computational Optimization and Applications (COA 2022); Computational Astrochemistry (CompAstro 2022); Computational methods for porous geomaterials (CompPor 2022); Computational Approaches for Smart, Conscious Cities (CASCC 2022); Cities, Technologies and Planning (CTP 2022); Digital Sustainability and Circular Economy (DiSCE 2022); Econometrics and Multidimensional Evaluation in Urban Environment (EMEUE 2022); Ethical AI applications for a human-centered cyber society (EthicAI 2022); Future Computing System Technologies and Applications (FiSTA 2022); Geographical Computing and Remote Sensing for Archaeology (GCRSArcheo 2022); Geodesign in Decision Making: meta planning and collaborative design for sustainable and inclusive development (GDM 2022); Geomatics in Agriculture and Forestry: new advances and perspectives (GeoForAgr 2022); Geographical Analysis, Urban Modeling, Spatial Statistics (Geog-An-Mod 2022); Geomatics for Resource Monitoring and Management (GRMM 2022); International Workshop on Information and Knowledge in the Internet of Things (IKIT 2022); 13th International Symposium on Software Quality (ISSQ 2022); Land Use monitoring for Sustanability (LUMS 2022); Machine Learning for Space and Earth Observation Data (MALSEOD 2022); Building multi-dimensional models for assessing complex environmental systems (MES 2022); MOdels and indicators for assessing and measuring the urban settlement deVElopment in the view of ZERO net land take by 2050 (MOVEto0 2022); Modelling Post-Covid cities (MPCC 2022); Ecosystem Services: nature’s contribution to people in practice. Assessment frameworks, models, mapping, and implications (NC2P 2022); New Mobility Choices For Sustainable and Alternative Scenarios (NEMOB 2022); 2nd Workshop on Privacy in the Cloud/Edge/IoT World (PCEIoT 2022); Psycho-Social Analysis of Sustainable Mobility in The Pre- and Post-Pandemic Phase (PSYCHE 2022); Processes, methods and tools towards RESilient cities and cultural heritage prone to SOD and ROD disasters (RES 2022); Scientific Computing Infrastructure (SCI 2022); Socio-Economic and Environmental Models for Land Use Management (SEMLUM 2022); 14th International Symposium on Software Engineering Processes and Applications (SEPA 2022); Ports of the future - smartness and sustainability (SmartPorts 2022); Smart Tourism (SmartTourism 2022); Sustainability Performance Assessment: models, approaches and applications toward interdisciplinary and integrated solutions (SPA 2022); Specifics of smart cities development in Europe (SPEED 2022); Smart and Sustainable Island Communities (SSIC 2022); Theoretical and Computational Chemistryand its Applications (TCCMA 2022); Transport Infrastructures for Smart Cities (TISC 2022); 14th International Workshop on Tools and Techniques in Software Development Process (TTSDP 2022); International Workshop on Urban Form Studies (UForm 2022); Urban Regeneration: Innovative Tools and Evaluation Model (URITEM 2022); International Workshop on Urban Space and Mobilities (USAM 2022); Virtual and Augmented Reality and Applications (VRA 2022); Advanced and Computational Methods for Earth Science Applications (WACM4ES 2022); Advanced Mathematics and Computing Methods in Complex Computational Systems (WAMCM 2022).

cs vs software engineering reddit: You Look Like a Thing and I Love You Janelle Shane, 2019-11-05 As heard on NPR's Science Friday, discover the book recommended by Malcolm Gladwell, Susan Cain, Daniel Pink, and Adam Grant: an accessible, informative, and hilarious introduction to the weird and wonderful world of artificial intelligence (Ryan North). You look like a thing and I love you is one of the best pickup lines ever . . . according to an artificial intelligence trained by scientist Janelle Shane, creator of the popular blog AI Weirdness. She creates silly AIs that learn how to name paint colors, create the best recipes, and even flirt (badly) with humans—all to understand the technology that governs so much of our daily lives. We rely on AI every day for recommendations, for translations, and to put cat ears on our selfie videos. We also trust AI with matters of life and death, on the road and in our hospitals. But how smart is AI really... and how does it solve problems, understand humans, and even drive self-driving cars? Shane delivers the answers to every AI question you've ever asked, and some you definitely haven't. Like, how can a computer design the perfect sandwich? What does robot-generated Harry Potter fan-fiction look like? And is the world's best Halloween costume really Vampire Hog Bride? In this smart, often hilarious introduction to the most interesting science of our time, Shane shows how these programs learn, fail, and adapt—and how they reflect the best and worst of humanity. You Look Like a Thing and I Love You is the perfect book for anyone curious about what the robots in our lives are thinking. I can't think of a better way to learn about artificial intelligence, and I've never had so much fun along the way. —Adam Grant, New York Times bestselling author of Originals

cs vs software engineering reddit: Women and Persona Performance Kim Barbour, 2023-07-25 This book works to unpack and explicate women’s personas. Drawing on global gender studies and feminist research, the author examines how ‘woman’ has been constructed socially, culturally, and politically throughout different historical periods and feminist movements. Case studies look at how women in different personal and professional settings construct, enact, and navigate their personas against a backdrop of shifting discourses on gender relations, continued patriarchal dominance, and western neoliberal capitalism. Chapters also delve into how women’s personas are constructed online through activism and community building. The author examines the diversity, flexibility, and slipperiness of the ways being a woman is experienced and strategically performed. This book will be useful for scholars and students in Gender Studies, Sociology, Psychology, and Media Studies.

cs vs software engineering reddit: Coders Clive Thompson, 2020-03-24 Facebook's algorithms shaping the news. Self-driving cars roaming the streets. Revolution on Twitter and romance on Tinder. We live in a world constructed of code--and coders are the ones who built it for us. Programmers shape our everyday behavior: When they make something easy to do, we do more of it. When they make it hard or impossible, we do less of it. From acclaimed tech writer Clive Thompson comes a brilliant anthropological reckoning with the most powerful tribe in the world today, computer programmers, in a book that interrogates who they are, how they think, what qualifies as greatness in their world, and what should give us pause. In pop culture and media, the people who create the code that rules our world are regularly portrayed in hackneyed, simplified terms, as ciphers in hoodies. Thompson goes far deeper, taking us close to some of the great programmers of our time, including the creators of Facebook's News Feed, Instagram, Google's cutting-edge AI, and more. Speaking to everyone from revered 10X elites to neophytes, back-end engineers and front-end designers, Thompson explores the distinctive psychology of this vocation--which combines a love of logic, an obsession with efficiency, the joy of puzzle-solving, and a superhuman tolerance for mind-bending frustration. Along the way, Coders ponders the morality and politics of code, including its implications for civic life and the economy and the major controversies of our era. In accessible, erudite prose, Thompson unpacks the surprising history of the field, beginning with the first coders -- brilliant and pioneering women, who, despite crafting some of the earliest personal computers and programming languages, were later written out of history. At the same time, the book deftly illustrates how programming has become a marvelous new art form--a source of delight and creativity, not merely danger. To get as close to his subject as possible, Thompson picks up the thread of his own long-abandoned coding skills as he reckons, in his signature, highly personal style, with what superb programming looks like. To understand the world today, we need to understand code and its consequences. With Coders, Thompson gives a definitive look into the heart of the machine.

cs vs software engineering reddit: *PREPARING FOR THE FUTURE: Transitioning into High-Growth Tech Careers* Sandeep Chavan, 2024-10-09 In today’s rapidly evolving technological landscape, the future of work is becoming more unpredictable, yet full of opportunities for those willing to adapt and learn. Preparing for the Future: Transitioning into High-Growth Tech Careers by Er. Sandeep Chavan, addresses this critical transition period, offering valuable insights into how the tech industry is shifting and what professionals need to do to stay ahead. The book dives into the growing influence of emerging technologies, such as artificial intelligence, quantum computing, augmented and virtual reality, and their significant impact on the job market. It explains how these cutting-edge technologies are not only reshaping current roles but also creating entirely new opportunities in fields that did not exist just a decade ago. Through well-researched insights and expert opinions, the book maps out a clear vision of where the future of work is heading and how individuals can prepare themselves for the challenges and opportunities of tomorrow. One of the book’s key strengths lies in its practical approach to addressing the needs of both new graduates and professionals looking to transition into tech. Whether you are a student exploring career paths or a seasoned worker contemplating a shift into IT, this book lays out step-by-step guides for navigating the changing job landscape. It covers everything from building an effective resume tailored for tech roles to acing interviews, understanding the importance of certifications, and leveraging online learning platforms for skill development. What sets Preparing for the Future apart is its comprehensive coverage of industry-specific trends. The book highlights not just the technical skills needed for success in tech but also the importance of soft skills such as communication, leadership, and networking. It emphasizes how a balanced combination of these skills can give candidates a competitive edge in a job market where collaboration across global teams is becoming increasingly important. Moreover, the book addresses the significant role of diversity and inclusion in the tech industry, explaining how a diverse workforce drives innovation and is key to the future success of organizations. It offers insights into initiatives aimed at bringing more underrepresented groups into the tech sector, an area of growing importance in India and globally. In addition to this, the book provides a global perspective, comparing India's IT sector to other major markets like the USA and China. It also discusses government policies that influence the sector and explores the rise of remote work and freelancing in the gig economy, which has opened up new avenues for tech professionals worldwide. In conclusion, Preparing for the Future: Transitioning into High-Growth Tech Careers is more than just a guide—it is a roadmap for anyone aspiring to thrive in the rapidly evolving world of technology. This book is a must-read for individuals ready to embrace change, build their skillsets, and secure their place in the tech-driven future.

cs vs software engineering reddit: The Laws of Trading Agustin Lebron, 2019-06-12 Every decision is a trade. Learn to think about the ones you should do — and the ones you shouldn’t. Trading books generally break down into two categories: the ones which claim to teach you how to make money trading, and the memoir-style books recounting scandals and bad behavior. But the former don't have profitable trades to teach; if they did they'd keep those trades to themselves. And the latter are frequently entertaining, but they don't leave you with much you can apply in your own life. The Laws of Trading is different. All of our relationships and decisions involve trading at some level. This is a book about decision-making through the lens of a professional prop trader. For years, behavioral and cognitive scientists have shown us how human decision-making is flawed and biased. But how do you learn to avoid these problems in day-to-day decisions where you have to react in real-time? What are the important things to think about and to act on? The world needs a book by a prop trader who has lived, breathed and taught trading for a living, drawing upon years of insights on the trading floor in real markets, good and bad, whether going sideways, crashing, or bubbling over. If you can master the decision-making skills needed to profitably trade in modern markets, you can master decision-making in all walks of life. This book will teach you exactly those skills. Introduces, develops, and applies one law per chapter, making it easy not only to remember useful concepts, but also to have them at the ready in any situation. Shows you how to find and think about the “special edge” of your organization, and yourself. Teaches you how to handle the interaction of people with artificially intelligent (AI) machines that make decisions, a skill that is rapidly becoming essential in the AI-driven economy of the future. Includes a bonus digital ancillary, an Excel spreadsheet with various worked examples that expand on the scenarios described in the book. Do you need to make rational decisions in a competitive environment? Almost everyone does. This book will teach you the tools that let you do your job better.

cs vs software engineering reddit: Python for Mechanical and Aerospace Engineering Alex Kenan, 2021-01-01 The traditional computer science courses for engineering focus on the fundamentals of programming without demonstrating the wide array of practical applications for fields outside of computer science. Thus, the mindset of “Java/Python is for computer science people or programmers, and MATLAB is for engineering” develops. MATLAB tends to dominate the engineering space because it is viewed as a batteries-included software kit that is focused on functional programming. Everything in MATLAB is some sort of array, and it lends itself to engineering integration with its toolkits like Simulink and other add-ins. The downside of MATLAB is that it is proprietary software, the license is expensive to purchase, and it is more limited than Python for doing tasks besides calculating or data capturing. This book is about the Python programming language. Specifically, it is about Python in the context of mechanical and aerospace engineering. Did you know that Python can be used to model a satellite orbiting the Earth? You can find the completed programs and a very helpful 595 page NSA Python tutorial at the book’s GitHub page at https://www.github.com/alexkenan/pymae. Read more about the book, including a sample part of Chapter 5, at https://pymae.github.io

cs vs software engineering reddit: Web Scraping with Python Ryan Mitchell, 2018-03-21 If programming is magic then web scraping is surely a form of wizardry. By writing a simple automated program, you can query web servers, request data, and parse it to extract the information you need. The expanded edition of this practical book not only introduces you web scraping, but also serves as a comprehensive guide to scraping almost every type of data from the modern web. Part I focuses on web scraping mechanics: using Python to request information from a web server, performing basic handling of the server's response, and interacting with sites in an automated fashion. Part II explores a variety of more specific tools and applications to fit any web scraping scenario you're likely to encounter. Parse complicated HTML pages Develop crawlers with the Scrapy framework Learn methods to store data you scrape Read and extract data from documents Clean and normalize badly formatted data Read and write natural languages Crawl through forms and logins Scrape JavaScript and crawl through APIs Use and write image-to-text software Avoid scraping traps and bot blockers Use scrapers to test your website

cs vs software engineering reddit: Security and Management and Wireless Networks Kevin Daimi, Hamid R. Arabnia, Leonidas Deligiannidis, 2025-04-26 This book constitutes the proceedings of the 23rd International Conference on Security and Management, SAM 2024, and the 23rd International Conference on Wireless Networks, ICWN 2024, held as part of the 2024 World Congress in Computer Science, Computer Engineering and Applied Computing, in Las Vegas, USA, during July 22 to July 25, 2024. For SAM 2024, 255 submissions have been received and 40 papers have been accepted for publication in these proceedings; the 12 papers included from IWCN 2024 have been carefully reviewed and selected from 66 submissions. They have been organized in topical sections as follows: Intrusion and attack detection: malware, malicious URL, phishing; security assessment and management + blockchain + use of artificial intelligence; cybersecurity and communications systems + cryptography and privacy; security and management + new methodologies and applications; wireless networks and mobile computing.

cs vs software engineering reddit: Mastering Robot design and programming Cybellium, Unleash Creativity and Ingenuity in Robotics In the realm of technology and automation, robots have become pivotal in reshaping industries and possibilities. Mastering Robot Design and Programming is your definitive guide to understanding and harnessing the potential of robotics, empowering you to create and program intelligent robots that push the boundaries of innovation and redefine the future. About the Book: As robotics technology evolves, the ability to design and program robots becomes increasingly crucial. Mastering Robot Design and Programming offers a comprehensive exploration of this dynamic field—an essential toolkit for engineers, enthusiasts, and innovators. This book caters to both newcomers and experienced learners aiming to excel in robot design, customization, and programming. Key Features: Robotics Fundamentals: Begin by understanding the core principles of robotics. Learn about robot components, kinematics, and dynamics that shape robot design. Custom Robot Design: Dive into custom robot design techniques. Explore methods for selecting mechanical parts, designing frames, and ensuring stability and mobility. Programming Basics: Grasp the art of robot programming. Understand how to write and upload code for motion control, sensors, and autonomous behaviors. Sensors and Perception: Explore sensors and perception systems for robots. Learn how to integrate cameras, LiDAR, proximity sensors, and other devices for accurate environment awareness. Autonomous Navigation: Understand the significance of autonomous navigation. Learn how to program robots to navigate through environments, avoid obstacles, and map surroundings. Human-Robot Interaction: Delve into human-robot interaction. Explore techniques for creating interfaces and behaviors that enable robots to collaborate with humans effectively. Robot Applications: Grasp real-world applications of robotics. From manufacturing to healthcare, discover the diverse applications of intelligent robots. Ethics and Safety: Gain insights into robotics ethics and safety considerations. Learn how to design robots that adhere to ethical standards and ensure safe operations. Why This Book Matters: In an age of innovation and automation, mastering robot design and programming offers a competitive edge. Mastering Robot Design and Programming empowers engineers, enthusiasts, and technology innovators to leverage robotics technology, enabling them to create intelligent robots that redefine industries and drive progress. Shape the Future of Automation: In the landscape of technology and automation, robots are at the forefront of reshaping industries and possibilities. Mastering Robot Design and Programming equips you with the knowledge needed to leverage robotics, enabling you to create and program intelligent machines that redefine innovation and open doors to new horizons.

Whether you're a seasoned robotics enthusiast or new to the world of robot design, this book will guide you in building a solid foundation for effective robot customization and programming. Your journey to mastering robot design and programming starts here. © 2023 Cybellium Ltd. All rights reserved. www.cybellium.com

Related to cs vs software engineering reddit

□□□□□□□cs□□□□□□□ - □□ □□□□□□□□□□CS□□□□□□□□□□□□□□□□□□□□ 1□□□□□□□□□□□□□□□□□□□□ 2□□□□□□□□□□□□□□□□□□□□□□ 3□□□□□

csgo□□□□□□□□□□□□□□□□□□□□□□□? - □□ □□□□□□□□□□□□□ESC□□□ □

□□□□□CS□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□CS□□□□□□1.6□□□□□□□□□□□□□□□□□□CS:GO□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

cs2□□□□□□□□□□□□□□□□? - □□ □□□□□□□ □□□ □□□□□ bug□□□□□□□□□□ □□□□□□□ CS□□□□□ □□□□□□□□□□□□□□□□□□□ - windowed -w 1920 -h 1080 □□□

CS:GO □□□□□□□□□□□□□□□□ - □□ CS:GO □□□□□□□□□□□□□□□□ □□□□ 2□CS2□ 77% □□□□

□□□□□□CS□□□□□□□□□□□□□□□□□CS□ □□□□□□CS□□□□□□□□□□□□□□□□□□□CS□ □□□□□□□□□□□□□□□□□□CS□□□□□□□□□□CS□□. □□□□□□□□□□□□□□□□□□□

□□□□□CS□□□rating□□□□□□□□□□□□2.0□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□CS□□□□□□□□FPS□□□□OW1□□□□□□□□□□□□□□□□□□□□□□□□□□□ [□

c# - Why am I getting error CS0246: The type or namespace name I am using Snarl C# API to send notifications to snarl. Now I have saved the content of above url in a file named SnarlNetwork.cs and the content of my test.cs file are: using

CS:GO □□□□□□□□□□□□□□□ - □□ CSGO□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□ □□□□□ □□□□□□□□□□□□□□□□□□ □□□ □□□□□□□□□□□5□60□□□□□□

c# - The annotation for nullable reference types should only be I have a console app to try out the C# 8 null reference types. Switched the project to build with lang ver C# 8. Then the following code results in a warning. class Program { static void Main(s

□□□□□□□cs□□□□□□□ - □□ □□□□□□□□□□CS□□□□□□□□□□□□□□□□□□□□ 1□□□□□□□□□□□□□□□□□□□□ 2□□□□□□□□□□□□□□□□□□□□□□ 3□□□□□

csgo□□□□□□□□□□□□□□□□□□□□□□□? - □□ □□□□□□□□□□□□□ESC□□□ □

□□□□□CS□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□CS□□□□□□1.6□□□□□□□□□□□□□□□□□□CS:GO□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

cs2□□□□□□□□□□□□□□□□? - □□ □□□□□□□ □□□ □□□□□ bug□□□□□□□□□□ □□□□□□□ CS□□□□□ □□□□□□□□□□□□□□□□□□□ - windowed -w 1920 -h 1080 □□□

CS:GO □□□□□□□□□□□□□□□□ - □□ CS:GO □□□□□□□□□□□□□□□□ □□□□ 2□CS2□ 77% □□□□

□□□□□□CS□□□□□□□□□□□□□□□□□CS□ □□□□□□CS□□□□□□□□□□□□□□□□□□□CS□ □□□□□□□□□□□□□□□□□□CS□□□□□□□□□□CS□□. □□□□□□□□□□□□□□□□□□□

□□□□□CS□□□rating□□□□□□□□□□□□2.0□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□CS□□□□□□□□FPS□□□□OW1□□□□□□□□□□□□□□□□□□□□□□□□□□□ [□

c# - Why am I getting error CS0246: The type or namespace name I am using Snarl C# API to send notifications to snarl. Now I have saved the content of above url in a file named SnarlNetwork.cs and the content of my test.cs file are: using

CS:GO □□□□□□□□□□□□□□□ - □□ CSGO□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□ □□□□□ □□□□□□□□□□□□□□□□□□ □□□ □□□□□□□□□□□5□60□□□□□□

c# - The annotation for nullable reference types should only be I have a console app to try out the C# 8 null reference types. Switched the project to build with lang ver C# 8. Then the following code results in a warning. class Program { static void Main(s

□□□□□□□cs□□□□□□□ - □□ □□□□□□□□□□CS□□□□□□□□□□□□□□□□□□□□ 1□□□□□□□□□□□□□□□□□□□□ 2□□□□□□□□□□□□□□□□□□□□□□ 3□□□□□

csgo□□□□□□□□□□□□□□□□□□□□□□□□? - □□ □□□□□□□□□□□□□□ESC□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□ □□
□□□□□CS□□□□□□□□□□□□□□□□□□□□□□□ □□□□□□□□□CS□□□□□□□1.6□□□□□□□□□□□□□□□□□□□CS:GO□□□□□□□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□□□
cs2□□□□□□□□□□□□□□□□□? - □□ □□□□□□□ □□□ □□□□□ bug□□□□□□□□□□ □□□□□□□□ CS□□□□□ □□□□□□□□□□□□□ □□□□□ - windowed -w 1920 -h 1080 □□□
CS:GO □□□□□□□□□□□□□□□□ - □□ CS:GO □□□□□□□□□□□□□□□□ □□□□ 2□CS2□ 77% □□□□
□□□□□□□CS□□□□□□□□□□□□□□□□□□CS□ □□□□□□□CS□□□□□□□□□□□□□□□□□□CS□ □□□□□□□□□□□□□□□□□□CS□□□ □□□□□□□□CS□□. □□□□□□□□□□□□□□□□□□□
□□□□□CS□□□rating□□□□□□□□□□□□2.0□□□□ □□□□□□□□□□□□□□□□□□□□□□□□□□□CS□□□□□□□□FPS□□□□OW1□ □□□□□□□□□□□□□□□□□□□□□□□□□□ [□
c# - Why am I getting error CS0246: The type or namespace name I am using Snarl C# API to send notifications to snarl. Now I have saved the content of above url in a file named SnarlNetwork.cs and the content of my test.cs file are: using
CS:GO □□□□□□□□□□□□□□□ - □□ CSGO□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ □□□ □□□□□ □□□□□□□□□□□□□□ □□ □□□ □□□□□□□□□□□□5□60□□□□□□□
c# - The annotation for nullable reference types should only be I have a console app to try out the C# 8 null reference types. Switched the project to build with lang ver C# 8. Then the following code results in a warning. class Program { static void Main(s

Related to cs vs software engineering reddit

Department of Computer Science and Software Engineering (Miami University5y) Today, computers are in everything, everywhere. This fact is making computer scientists and software engineers indispensable. They are the ones leading the way in developing the next generation
Department of Computer Science and Software Engineering (Miami University5y) Today, computers are in everything, everywhere. This fact is making computer scientists and software engineers indispensable. They are the ones leading the way in developing the next generation
What is Software Engineering? (Michigan Technological University3y) Software engineering is the branch of computer science that deals with the design, development, testing, and maintenance of software applications. Software engineers apply engineering principles and
What is Software Engineering? (Michigan Technological University3y) Software engineering is the branch of computer science that deals with the design, development, testing, and maintenance of software applications. Software engineers apply engineering principles and

Back to Home: https://test.murphyjewelers.com