

Arnav Bhalgat

April 23, 2024

## **UC CS Discourse Community**

### **Methods for Analysis:**

In order to understand the community of computer science (CS) students at the University of Cincinnati (UC), this essay will employ a multifaceted approach drawing from a diverse array of resources. Interviews with students and professors, will provide invaluable insights into the goals and communication methods of the CS community. Additionally, scholarly sources such as Jessen Havill's *"Discovering Computer Science"* and *"The Concept of Discourse Community"* by John Swales, will serve as pillars of theoretical framework and foundational knowledge. Furthermore, Gabriella Barcinas' research paper offers a framework for understanding the UC Computer Science community's structure and communication methods, particularly through platforms like *GitHub*. Additionally, a genre analysis of communication on *GitHub* will unveil the community's specific lexis and modes of interaction. By incorporating insights from the *UC Computer Science Department website*, we delve into the diverse coursework and career opportunities within the community. This eclectic blend of qualitative and quantitative methods ensures a nuanced exploration of the vibrant and ever-evolving Computer Science community at UC.

### **Introduction:**

In the heart of bustling campus life at the University of Cincinnati, there exists a world within College of Engineering and Applied Sciences — a community veiled in the enigmatic glow of screens and the rhythmic hum of keyboards. Welcome to the universe of computer science,

**ARNAV A BHALGAT**

where we play with ones and zeros to unlock the secrets of our digital age and liberate humanity of its problems. Seems a lot, doesn't it? But fear not, here's our roadmap for achieving this vision. This vibrant ecosystem aligns the literacy of computers with the literacy of CS students, and the quest for knowledge is a never-ending expedition. As eminent futurist and philosopher Alvin Toffler once said, “the illiterate of the 21st century will not be those who cannot read and write, but those who cannot learn, unlearn, and relearn.”

But what sets this community apart? What makes it tick, thrive, and evolve in a world where innovation is the currency of progress? The answer lies in its very essence – a deep-seated passion for computers and the code that propels them forward. This inherent love for computers and the driving force of code is precisely why I embraced computer science as my major and chose to join the community. Here, ideas are not merely shared but dissected, debated, and refined through a continuous exchange of thoughts and insights. Whether you're a seasoned programmer or a curious outsider peeking into this world of wonder, I found the UC computer science community welcoming me with open arms. So, steel yourself for a voyage through the intriguing depths of this community. Get ready to unravel the mysteries of the digital realm and discover the boundless possibilities that await within the computer science community here at University of Cincinnati.

### **Shared Goals and Interests:**

Picture yourself amidst a labyrinth of code, where every line tells a story, and every bug presents a challenge waiting to be conquered. Here, in the halls of UC's Computer Science Department, students gather not just to study, but to immerse themselves in a discourse community unlike any other. It's a place where curiosity reigns supreme, where the pursuit of

understanding is both a solitary journey and a collective endeavor. As I navigated the maze of algorithms with utter love for mathematics, I was welcomed and greeted by fellow tech geeks eager to share their wisdom and experiences. Ice-breaker communications and topic introductions kicked off my journey where I bonded, felt at home, and made friends. It's like creating a tight-knit group, sort of like when you use *nested loops* and *functions* in coding to group things together. Swales says, “a discourse community can be identified based on the shared goals and interests that are exercised by each member of the community” (220). But what are the goals of this community, you may ask? It is, above all, a platform for collaboration, networking and most importantly growth.

Within this dynamic community, learning fundamentals of computer science takes on a whole new dimension. Unlike other majors, where independent study may suffice, the CS community thrives on collaboration and shared exploration. Through coding prompts and problem-solving labs, we sharpen our skills and expand our horizons to find the "desired solution", all while fostering a sense of brotherhood that sustains us through the challenges ahead. Nobody codes alone these days, coding in isolation is a relic of the past! But collaboration isn't just about sharing ideas; it's about pushing boundaries and expanding horizons. As professor Hawkins, an esteemed member of this community, puts it, “With more minds comes more ideas.” And perhaps most importantly, collaboration breeds confidence. From troubleshooting code to presenting projects, students learn to navigate “the complexities of computer science” with newfound ease and proficiency. As I reflect and compare my experiences, “Teamwork brings out the best of any student, but CS students need it the most.” And in a field where continuous learning is the norm, the support and inspiration of the CS community serve as invaluable resources on the journey toward mastery.

I fondly recall immersing myself in the atmosphere of the MakeUC hackathon, where I ventured into the world of web development amidst a backdrop of friendly competition. It

was reminiscent of the spirited challenges inherent in the world of computer science. Whether through hackathons, structured curriculum, informal discussions, or webinars featuring industry professionals — members are constantly exposed to new concepts, refining their interests, and exposing them to real-world practices and industry standards. Indeed, the collective wisdom of the CS community fosters a culture of innovation where students learn searching for appropriate resources to support their projects (Hawkins). Do you know CS curriculum cannot be learnt by heart, even the most seasoned coders rely on *GitHub*, Language Documentation, and AI tools to navigate the intricacies of syntax!

Ultimately, the purpose of our community is clear: to cultivate a new generation of computer scientists equipped not only with technical prowess but also with the collaborative spirit and critical thinking skills necessary to tackle the real-world problems. Through collaboration, exposure, and a shared commitment to excellence, we pave the way for a brighter, more innovative future in computer science.

### **Lexis and Literacy**

Course learning in this major involves a specific comprehension of syntax which revolves around algorithms and is a distinct component in programming languages and allows humans to clearly communicate with computers (Havil 4). In this community of computer science students' communication transcends mere words; it's a language unto itself, spoken fluently by those immersed in its intricacies. Professor William Hawkins underscores the importance of active listening as a cornerstone of success in any community. "It's the first step towards feeling included, where everyone's voice is heard, respected, and valued," he passionately emphasizes. In this environment, communication takes on an argumentative, provocative tone, with members engaging in "pedantic conversations" (Hawkins) to push and prove the

efficacy of their logic. For first year students, understanding the customs and rituals of the community is as important as participating in it; focusing on active listening can yield valuable insights and pave the way for seamless integration into the vibrant discourse.

As John Swales posits, communication within a discourse community is facilitated by a unique set of "lexical items" understood primarily by its members (22). Within the UC Computer Science community, this lexicon is rich and diverse, comprising programming terminology that serves as the bedrock of communication. Swales explicitly states that "information technology discourse communities" obtain a range of terminology in order to achieve their collective goals, which in this case is knowing the wide range of operations necessary to form a program, dependent on the computer language one is using (222).

Me: "Hey, I've just pushed the latest changes to our project repository on GitHub. Can you pull them and troubleshoot the bugs?"

My Friend: "Sure thing! I'll pull the changes and review them. By the way, did you remember to commit the latest updates before pushing?"

Me: "Yes, I committed them before pushing. Everything should be up to date now."

*A typical conversation between two computer science students, they discuss the process of pushing and pulling changes on GitHub, a popular collaboration platform for developers.*

Terms such as "push," "pull," "commit," and *backtick* ( ` ) used above become shorthand for complex programming concepts, indicative of one's CS background. Similarly, phrases like "if-else statements," "for loops," and "while loops" serve as the building blocks of algorithmic thinking, akin to the grammar rules in spoken language. Moreover, the usage of "functions" and "methods" can be likened to verbs in a sentence, where they perform specific actions or operations. Just as punctuation marks aid in structuring sentences, symbols like round

brackets and curly braces delineate the structure and flow of code. In this way, the language of programming mirrors the intricacies of human language, with each term and symbol contributing to the coherent expression of ideas within the CS community. Yet, communication in the UC Computer Science community extends beyond the confines of face-to-face interactions. As Devansh, a fifth-year Mechanical Engineering student, observes, communication is predominantly short, to-the-point and online. Unlike the regimented meetings of his engineering counterparts, CS students majorly rely on virtual collaboration platforms to discuss and compile their work. This widespread communication fosters a culture of development, where ideas are shared and refined across vast digital landscapes with no bounds.

### **Genre Analysis:**

In the electrifying arena of computer science discourse community where communication transcends traditional confines, reaching its apex in the virtual corridors of *GitHub*. Here CS students converge to showcase, troubleshoot, and innovate with unparalleled efficiency and precision. *GitHub*, akin to *LinkedIn*, serves as the nexus of communicative exchange within the CS community. At its core, *GitHub* embodies the essence of collaboration, enabling students to engage in iterative problem-solving and collective innovation with seamless fluidity. Through a myriad of features such as "opening issues," "creating branches," and "forming pull requests," members navigate the intricate web of code changes and discussions with utmost clarity and purpose.

The most striking aspect of GitHub's communicative landscape lies in its objective-driven approach. Here, communication is stripped of embellishment and narrative, focusing solely on the task at hand – addressing and resolving coding issues. The genre, therefore, is one of

factual precision and critical analysis, devoid of extraneous details or anecdotes. Within this virtual arena, each interaction serves a specific purpose – to propel projects forward and unlock new realms of possibility. Moreover, GitHub's expansive reach extends beyond the confines of individual projects, offering a platform for knowledge dissemination and community enrichment. Through the "Explore" tab, students can glean insights from the research and trial-and-error processes of their peers, fostering a culture of shared learning and collective advancement.

In essence, GitHub emerges as more than just a repository of code; it embodies the very spirit of collaboration and innovation that defines the CS community at the University of Cincinnati. Through its seamless integration of communication and problem-solving, GitHub empowers students to push the boundaries of possibility and shape the future of technology with unwavering determination and collective ingenuity.

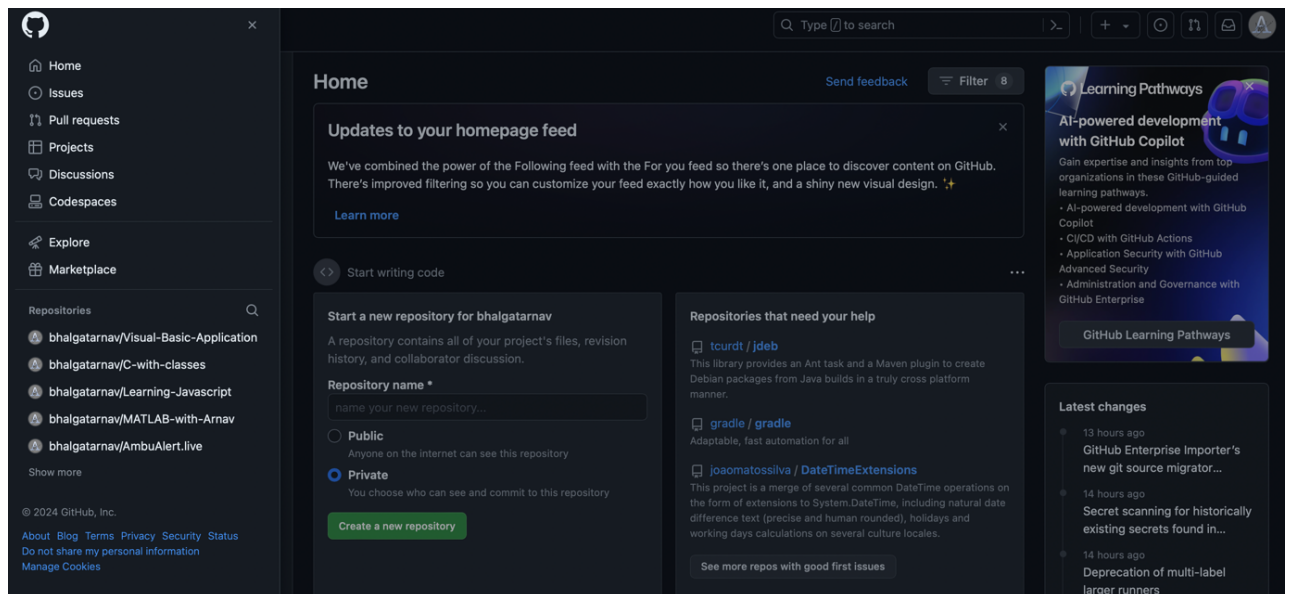


Figure: Overview of the GitHub website from the home page

**Conclusion:**

In conclusion, immersing oneself in the pulsating discourse community of computer science at the University of Cincinnati reveals a landscape brimming with innovation, collaboration, and endless opportunities. Through insightful interviews and meticulous analysis of communication, we've unearthed the shared aspirations and values that drive this exhilarating community forward. From nurturing critical thinking abilities to equipping students for real-world challenges, the CS community at UC serves as a crucible for budding scientists and trailblazers. As the demand for tech expertise continues to surge, so does the allure of exploring the myriad career avenues within computer science. This research endeavors to ignite the imagination and embolden young undergraduates to embark on their own odyssey within this thriving discourse community, where the horizon is boundless and the possibilities, truly limitless.

## Works Cited

Arnav Bhalgat, and Professor William Hawkins. Personal Interview (8 April 2024).

Arnav Bhalgat, and Devansh Saxena. Personal Interview (11 April 2024).

Barcinas, Gabriella. "Computer Science as a Discourse Community | FYC Journal."

Computer Science as a Discourse Community, 30 Nov. 2020, [fycjournal.ucdavis.edu/wp-content/uploads/2021/01/Computer-Science-as-a-Discourse-Community.pdf](http://fycjournal.ucdavis.edu/wp-content/uploads/2021/01/Computer-Science-as-a-Discourse-Community.pdf).

Swales, John. "The Concept of Discourse Community." *Genre Analysis: English in Academic and Research Settings*. Boston: Cambridge UP, 1990. 21-32. Print.

"What Is Computation?" *Discovering Computer Science*, Jessen Havill, CRC Press, 2015, pp. 1–5, [learning.oreilly.com/library/view/discovering-computer-science/9781482254167/](http://learning.oreilly.com/library/view/discovering-computer-science/9781482254167/).