John Carl E. Bautista
BSIT - 1 (MW 1:30 - 3:00)

PORTFOLIO #1
Computer Science as Discipline, Computing Disciplines and Majors

Computer Science as Discipline

- When I read the articles/works of Hopcroft, Steedman, Denning, Denning & Martell, Wing, and Tedre, I gained a deeper understanding of how computer science has developed into a discipline and why it matters to study it as one. Hopcroft's Turing Award Lecture highlighted how the field established itself by building strong theoretical foundations such as automata and complexity theory, and how collaboration and institutional support played a key role in its growth. Denning's Great Principles of Computing emphasized that computer science is united by broad principles, like communication, computation, and evaluation, that remain important even as technologies change. Steedman's reflections on related fields reminded me that without unity and structure, a discipline struggles to be taken seriously, which made me appreciate how computer science has worked to define its scope clearly. Denning & Martell reinforced this by showing that great principles can also be used to connect theory with practice in education, helping students see computing as more than just coding. Wing's article on Computational Thinking expanded my perspective by showing that the mindset of computer science, problem decomposition, abstraction, and algorithmic thinking, can be applied by everyone, not just computer scientists. Finally, Tedre's work on teaching the philosophy of computer science added another layer, making me realize that the field also needs to reflect on its philosophical, methodological, and scientific foundations to strengthen its identity and guide its future.

I mostly chose works that come from the 2000s, a time when computer science was rapidly expanding beyond its early theoretical foundations into a discipline that influences education, philosophy, and even everyday problem-solving. This timing is important because it reflects how the field matured: by then, computer science was no longer just about building machines or writing efficient code but about defining its principles, teaching its philosophy, and explaining its relevance to other domains.

Taken together, these readings show that computer science is not only about building software or hardware, it is a true discipline because it has theories, principles, philosophies, and methods that define it. At the same time, it remains powerful because it shapes the way people think and solve problems across all areas of life. The challenge for students like me is to study beyond just coding or tools, to engage with the theories and principles that give the field depth, while also being aware about its limits.

Keypoints -

-Hopcroft: Computer science was built on strong theories like automata and complexity.

-Denning: The field is tied together by great principles that outlast new technologies.

-Steedman: Disciplines need unity and structure to gain respect.

-Denning & Martell: Principles connect theory with practice in computing education.

-Wing: Computational thinking is a problem-solving skill for everyone, not just programmers.

-Tedre: Philosophy helps define the foundations and identity of computer science.

5 Computing Disciplines and Majors

When we study computing as a discipline, it becomes clear that it is not a single path but a family of majors that work together to build the digital world we live in. Computer Science, Computer Engineering, Information Systems, Information Technology, and Software Engineering each bring a different focus, but they are deeply connected. The theories and algorithms of computer science lay the foundation for how we understand computation, while computer engineering makes those ideas possible in physical form through hardware and integrated systems. On top of this foundation, software engineering provides the methods to design and maintain complex applications that people rely on every day, from mobile apps to medical systems. These systems do not exist in isolation, information systems ensure that technology is used effectively within organizations, turning raw data into meaningful knowledge for decision-making. Information technology supports all of this by maintaining networks, securing data, and making sure the entire infrastructure runs reliably.

What stands out to me is that these majors are not equal in visibility or popularity. Computer science tends to get most of the attention because of things like AI, while fields like information systems or IT are often seen as less “prestigious,” even though businesses and institutions would collapse without them. Software engineering promises stable careers but also carries the heavy responsibility of avoiding errors that could cost money, or lives. Computer engineering often requires expensive labs and equipment, which not all schools can provide. This shows me that while the five majors form an interconnected ecosystem, they also face real challenges in how they are valued, taught, and supported.

For me as an IT student, understanding these five majors as part of one discipline is not about just being educated, it’s about being realistic. Computing is powerful, but it also comes with limits and responsibilities. In IT especially, we often deal with the practical side of technology: making sure systems are secure, networks stay up, and infrastructure actually works when people need it. It’s a role that is sometimes overlooked compared to the attention given to fields like computer science or AI, but without IT, all the theories and applications would never make it into daily use. That perspective shapes how I see the whole discipline: every major has its struggles, whether it’s keeping up with rapid change, fighting stereotypes, or carrying the weight of building systems people depend on.

keypoints

-The five computing majors are interconnected, but not equally valued or visible.

-Computer science often overshadows the others, though IS and IT are critical in practice.

-Each field faces real challenges: resource needs, societal perceptions, and responsibility for failures.

-Computing is not only about opportunities but also about responsibilities and limits.

-As students, we must see the discipline as a serious, demanding field that shapes society.

ABOUT ME

My Name is John Carl E. Bautista

Most of my elementary and junior high years were spent at Liceo Del Verbo Divino (LVD) in Leyte, which is also under the SVD. I then transferred to Eastern Visayas State University for my senior high school. After that, I enrolled at the University of San Carlos because I heard about the quality of education offered here.

As to why I chose BSIT, it was actually not my first choice. I initially wanted to take Architecture, but I realized it was not practical, so I decided not to pursue it. Aside from that, I chose BSIT because of the high demand for IT professionals, the versatile career paths it offers, and the opportunity to work right away after graduating.

references:

Denning, P. J. (2003). Great principles of computing. *Communications of the ACM, 46*(11), 15–20. https://doi.org/10.1145/948383.948400

Hopcroft, J. E. (1987). Computer science: The emergence of a discipline. *Communications of the ACM, 30*(3), 198–202. https://doi.org/10.1145/214748.315390

Steedman, M. (2008). Last words: On becoming a discipline. *Computational Linguistics, 34*(1), 137–144. https://doi.org/10.1162/coli.2008.34.1.137

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Tedre, M. (2007). Know your discipline: Teaching the philosophy of computer science. *Journal of Information Technology Education, 6*(1), 105–122.https://www.learntechlib.org/d/111410/

Britannica. (n.d.). *Computer science: The study of computers and computing*. In *Britannica.com*. Retrieved from https://www.britannica.com/science/computer-science

Connolly, A., et al. (2017). *Computing disciplines: A quick guide for prospective students and career advisors*. McGill University Career Planning Service. Retrieved from https://www.mcgill.ca/caps/computing-disciplines-quick-guide-prospective-students-and-career-advisors

OpenStax. (2024). *Introduction to Computer Science*. OpenStax. https://openstax.org/books/introduction-computer-science

TechJunkie. (n.d.). *An overview of computing disciplines*. TechJunkie. Retrieved from https://www.techjunkie.com/overview-computing-disciplines/

Wikipedia contributors. (n.d.). *Computer engineering*. In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Computer_engineering