REPORT OF THE LSA COMPUTING EDUCATION TASK FORCE *

August E. (Gus) Evrard (co-chair)
Arthur F. Thurnau Professor
Departments of Physics and Astronomy
College of LSA
evrard@umich.edu

Mark J. Guzdial (co-chair)
Professor
Department of Electrical Engineering & Computer Science
College of Engineering & School of Information
mjguz@umich.edu

Elizabeth Bruch
Associate Professor
Department of Sociology, Complex Systems
College of LSA
ebruch@umich.edu

Nick Henriksen
Associate Professor of Spanish Linguistics
Department of Romance Languages
College of LSA
nhenriks@umich.edu

Amir Kamil
Lecturer IV
Department of Electrical Engineering & Computer Science
College of Engineering
akamil@umich.edu

Richard Lewis
John R. Anderson Collegiate Professor
Departments of Psychology, Linguistics and Cognitive Science
College of LSA
rickl@umich.edu

Annette Ostling
Associate Professor
Department of Ecology and Evolutionary Biology
College of LSA
aostling@umich.edu

Sarita Schoenbeck
Associate Professor of Information
School of Information
yardi@umich.edu

M. Remi Yergeau
Associate Professor
Department of English Language and Literature
Digital Studies Institute
College of LSA
myergeau@umich.edu

*https://sites.lsa.umich.edu/computingedu
EXECUTIVE SUMMARY

In September 2020, the Dean’s Office of the College of Literature, Science and the Arts (LSA), convened a task force charged with evaluating the state of computing education in LSA, identifying unmet needs, and creating recommendations along with an implementation plan\(^1\). This report presents our response to that charge. We outline here our key recommendations.

**Cultivate a broad, coordinated vision for computational and digital (C&D) education across LSA.** Science and society have thrived from advances in computing, but networked digital tools can also be weaponized and degrade society. LSA should expand its early investments in digital studies and quantitative social science to grow an educational culture in which every LSA student can learn how to leverage computing for the good of society and science. LSA, internally and in partnership with other schools, should work to create new computing education opportunities and credentials for its students across the full range of creative and liberal arts and sciences. Students should leave LSA feeling confident in their abilities to design and employ computational methods and digital tools both effectively and critically, thereby expressing LSA’s mission “to foster the next generation of rigorous and empathetic thinkers, creators, and contributors to the state of Michigan, the nation, and the world” [1].

**Develop New Curricular Opportunities and Degree Pathways.** From the late 1980’s until very recently, an LSA student interested in computing generally had one degree path to pursue – the bachelor of science in Computer Science. LSA should expand existing minor programs in Digital Studies and Quantitative Methods in Social Science into major programs, and encourage new computational degrees across the C&D spectrum, including new science degrees that infuse data science and computer science capabilities with domain knowledge as well as new humanities degrees that use digital tools to expand human creativity and expression and to support a more just, equitable and truthful society. Faculty should be encouraged to collectively develop new courses in support of these programs, and departments encouraged to grow and recognize a variety of skill enhancement opportunities and competency credentials, including online learning experiences.

**Establish a Home for C&D Education in the College.** Computer science education for LSA students has effectively been outsourced to the College of Engineering for decades. LSA should build its own home for computational and digital education that would serve to enact the above recommendations and stand at the center of C&D education activities for the College. We outline several potential options for a home, including an Institute or Program in Computing for Society and Science.

**Create a leadership position for Computing, Information and Data in the College.** Computing cuts across the traditional academic divisions, and investments in cyberinfrastructure and digital tools are core elements that support the College’s missions of research and creative arts, teaching, and public service. LSA should create a leadership position within the Dean’s Office for a faculty member charged with coordinating and implementing policies and practices in computing, information and data management.

---

**N.B.** The work of the task force has been conducted exclusively using digital tools, not as a demonstration of principle but out of necessity due to the Covid-19 pandemic. We deeply thank all members of the Michigan community who shared time and energy with the committee during this difficult period, and offer condolences to those who suffered personal losses because of this crisis.

---

\(^1\)The implementation plan, part of the original charge, was subsequently dropped.
Contents

1 Background and Charge .......................... 5
  1.1 More than just Computer Science: Core Themes for a Digital Education ............ 5
  1.2 Computing for the Sciences: Computational Thinking, Model Thinking and Beyond ............. 7
    1.2.1 Recent developments within LSA ......................................................... 8
    1.2.2 From Digital to Quantum Computing ..................................................... 8
    1.2.3 Forms of Intelligence and Cognitive Science .......................................... 8
  1.3 Computing for Society: Critical Studies of Identity, Expression and Decision-making Algorithms ...... 8
    1.3.1 Computing for Expression: New Forms of Media and Property ...................... 9
  1.4 Increasing Demand for, and Limits to, Participation in LSA Computing Education ............. 10
  1.5 The University as a Leader in Statewide Computing Education ............................. 11
  1.6 Computing and the Environment ............................................................... 12

2 Findings ............................................. 13
  2.1 Summary of Findings ................................................................. 13
  2.2 Interviews and Surveys ............................................................ 13
    2.2.1 Early Assessment from TF Member Experiences ......................................... 14
    2.2.2 Initial Interviews ................................................................. 14
    2.2.3 Survey of Computing Educators in LSA ............................................... 15
    2.2.4 Discussions with LSA Students ......................................................... 23
    2.2.5 Interviews with Computing Education Informants in LSA ......................... 23
  2.3 Searches of the C&D Curricular Landscape in LSA ........................................ 25
    2.3.1 Brief Summary of Curricular Findings .................................................. 25
    2.3.2 Search process ........................................................................... 25
    2.3.3 Course Description Search, AY 2019-2020 ............................................. 26
    2.3.4 Syllabus Search, Fall 2015 to Winter 2020 ........................................... 29
    2.3.5 C&D Courses and College Requirements ............................................. 29
    2.3.6 The Curious Case of EECS 101 ......................................................... 30
    2.3.7 Who is engaged in Computer Science and Data Science degrees? .................. 31
  2.4 Computing Education Initiatives and Peer institutions ........................................... 32
    2.4.1 Novel and Interesting Programs ......................................................... 32
    2.4.2 Comparison with Benchmark Peer Institutions ......................................... 32

3 Recommendations ...................................... 33
  3.1 A Coordinated Vision for Computational and Digital Education ............................. 33
  3.2 Curricular Enhancements ...................................................................... 34
  3.3 A Home for Computational and Digital Education in the College ......................... 35
  3.4 Dean’s Office Leadership ................................................................... 36

A Charge for an LSA Computing Education Task Force ........................................... 38
B Searches of Course Descriptions and Syllabi: Digital and Computational Terms

C LSA Faculty Teaching Computing Survey

C.1 Any additional thoughts on “Computing for Discovery” ........................................ 41
C.2 Any additional thoughts on “Computing for Expression” ................................. 42
C.3 Any additional thoughts on “Critical Computing” ................................................. 43
C.4 What’s missing from these definitions ................................................................. 45
C.5 What other curriculum options should we make available? ............................... 46
C.6 What is working in LSA to provide computing education to our undergraduates? 48
C.7 What do we need to fix to provide LS&A undergraduates with the computing education they need? ............................................. 49
C.8 What do we need to add to LS&A to provide undergraduates with the computing education they need (e.g., new courses, programs, departments)? ......................................... 52
C.9 What else should we be thinking about? .............................................................. 55

D Novel and Interesting Computing Education Programs ........................................ 57

E Comparison with Peer Institutions .......................................................................... 64

F Detailed Interview Notes with Computing Education Leadership in LSA ................ 67
1 Background and Charge

Computing technologies have transformed the human experience, enabling new modes of scientific discovery and new forms of expression. As technological progress drove exponential reductions in the costs of communication, data processing and data storage over the past half-century, a glut of networked capacity emerged. Outfitted with sophisticated, real-time algorithms, the world-wide web now supports the familiar, ad-based economy dominated by a handful of multinational corporations such as Facebook, Apple, Amazon, Microsoft and Google.

The digitization of society is mirrored in the academy. In a taste from the world of design and architecture, Ellie Abrons and collaborators declare in *Becoming Digital*,

To become digital is to be situated in a context where everything from screen to stone exists as both data and matter, where habits of mind forged within the digital environment are constantly transferred to the analog world, and vice versa [2].

As a premier public institution in the liberal arts tradition, one with a long and rich history in computer communications and computer science, the College of Literature, Science and Arts (LSA) deserves to remain at the forefront of computational science and digital studies. Its graduates should operate in a digital-first world effectively, with purpose grounded in the liberal arts traditions of progressive enlightenment and positive advancement of society and science.

In Fall 2020, LSA Associate Dean of Undergraduate Education, Tim McKay, convened a small *ad hoc* Task Force (TF) to address the state of computing education in the College.² The history of the LSA computer science degree spans nearly seven decades, with origins as a pioneering interdisciplinary program. In 1965, one year after a university-led proposal to develop the MERIT network³, the Department of Computer and Communication Sciences began offering an undergraduate concentration in Communication Sciences. In a 2001 interview, Bernie Galler, department chair in the early 1970’s, explained the broad perspective for the degree,

We didn’t want to call it Computer Science because it wasn’t a standard computer science department — it had a great deal of interdisciplinary activity. We had a course on biological systems as information processors, and a course on behavioral systems as information processors, along with programming and other things normally found in computer science [3].

By the mid-1980’s, a growing number of computer-related programs across the university led to a centralization of computing instruction in a new Department of Electrical Engineering and Computer Science (EECS) located in the College of Engineering (CoE). The Computer Science (CS) degree, however, remained an LSA credential, until CoE developed its own option in 2001. This situation, enacted almost forty years ago, remains the current state of affairs.

The tectonic shifts that have occurred in computing and digital citizenry over that time prompt a re-evaluation of LSA’s curricular offerings in these areas. The essential charge to the committee was to undertake the following.

1. Explore what computing is being taught to LSA students, and to whom, and identify unmet needs.
2. Develop plans to provide computing education for LSA students.
3. Draft recommendations for implementation over a three year period.

Full text of the original charge is given in Appendix A. After the TF was convened, the implementation plan was subsequently removed and so is not included as part of this report. This report reviews our findings and summarizes our recommendations.

1.1 More than just Computer Science: Core Themes for a Digital Education

The faculty members of the task force come from a range of backgrounds. While two members are rooted in computer science, one a computing education specialist, the rest span the range of the College’s historic liberal arts reach. One has multiple appointments involving social science and complex systems, three are natural scientists (in evolutionary biology, physics, and psychology), one is a digital humanist, another a more analog humanist, and one is an information scientist whose research focuses on social computing from a humanistic perspective. This breadth led us to interpret the idea of “computing education” broadly⁴, spanning a spectrum from digital studies in the humanities to computational studies in the sciences, as illustrated in Figure 1.

²Hereafter, unless otherwise specified, “the College” refers to the College of Literature, Science and Arts at the University of Michigan, Ann Arbor.
³The region’s first inter-university network would help spawn the internet. It continues operating at merit.edu.
⁴Expanded definitions of computing can be found at the Task Force website.
Figure 1: Illustration of the Task Force’s vision for computing education in the liberal arts, represented as a digital to computational spectrum. Humanists resonate with the term digital and natural scientists with computational; social scientists couple to both. The three core themes span this space, as reflected by our faculty survey (§2). Computing for Discovery applies model thinking and statistical techniques to interpret data, driving new science. Computing for Justice used multiple modalities to critique the impact of social media and other information technologies on society and the human condition. Computing for Expression uses computational methods in a multitude of creative ways, thereby spanning the entire dimension, from new forms of media and games to scientific visualization.

Early TF discussions produced consensus themes that we used to frame our findings and perspectives of computing education in the College.

Computing for Discovery. Computing greatly facilitates discovery across the disciplines. By encapsulating automation, computing enables the study of rich, complex systems in terms of non-linear system models composed of many interacting components. Infused with empirical data, such models enable scientists to reveal and refine theoretical principles. Models also form the basis of predictive applications based on simulation, and large data volumes support predictive application using machine learning (ML) and artificial intelligence (AI) methods. Increasingly, advanced computing capabilities and data science principles are driving transformational research discoveries in the natural, social, and economic sciences.

Computing for Expression. The computer has been called humanity’s first meta-medium: the first medium that could be any other medium (text, photography, movies, etc.) and make it interactive. Computing can be used to communicate and engage with others in ways that we couldn’t previously, which might include use and production of apps, virtual reality environments and games, social media, and the creation of new models that can be realized via simulation. Blockchain technologies facilitate new forms of expression in digital production, including cryptocurrencies, and digital rights management.

Computing for Justice, or Critical Computing. Now pervasive in our everyday lives, computers have immense cultural, social, and political influence. It is therefore imperative to ask who is supported by computing, who is oppressed by it, and how these outcomes impact the human experience. We should ask whether inequities that are reinforced or magnified by computing technologies are a justified consequence, or signal a need for technological refusal. We can create alternative models of, and infrastructures for, computing technologies that further welfare, justice, and equity [4].
Along with the above core themes, we highlight a fourth opportunity that builds on LSA’s humanistic strength.

**History of Computing Technologies.** Mechanical means of computing existed in ancient times, and the history of digital computing is nearing a century old. Humanity’s drive to understand and shape its condition is a story woven with digital and computational threads. Scholars in LSA, concerned with context, realize that technology is not separate from its development, including who developed it, and for what purpose and use. A study of the history of digital technologies deserves to be a cross-cutting element of LSA computing education. The fact that members of the Michigan community have made significant contributions to the theory of computing, the development of computing technologies, digital culture, digital curation, computational science, collaboratories and cyberinfrastructure could be used to situate and inspire students.

1.2 Computing for the Sciences: Computational Thinking, Model Thinking and Beyond

NASA’s human computers of the 1960’s, a crew of mostly women, performed approximate difference calculations by hand to steer spacecraft trajectories. Some of these early calculators went on to program digital computers to accomplish this task, at speeds thousands of times faster than humans. Many of these transitional computers were women of color, including Katherine Johnson and Mary Jackson, whose efforts the agency has recently come to recognize [5].

This example from spaceflight engineering exhibits modalities of computational thinking and model thinking, overlapping concepts grounded in the idea of abstraction, the ability to identify core elements of a system, characterize their interactions, and develop frameworks aimed at specific outcomes. To reach the Moon, the spacecraft is modeled as a point mass in a complex gravitational field of the Earth, Moon and Sun. Steering it requires burning its engines at precise times, with specific orientations, thrusts and duration. These features, the model’s degrees of freedom, are all computable given initial and final desired states and the specified gravitational field.

Over a decade ago, Jeannette Wing, Avanessians Director of the Data Science Institute and Professor of Computer Science at Columbia University, along with Jan Cuny of the National Science Foundation and Larry Snyder of the University of Washington, coined the following definition.

Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent [6].

Model thinking allows the formulation of problems in computable terms. As framed by Scott Page, the John Seely Brown Distinguished University Professor of Complexity, Social Science, and Management at Michigan, model thinking is a formal mathematical representation that can be applied to or calibrated to fit data [7]. And data is increasingly available from a growing variety of sources.

Essentially all science disciplines have been transformed by the so-called fourth paradigm [8] of science. More than a decade ago, the late Jim Grey of Microsoft Research framed knowledge creation in some science discipline, X, as a virtuous coupling of empirical data collection (X-Informatics) with first-principles modeling (Computational-X). For example, if you ask a random cosmologist, “What is a galaxy?”, two relevant answers are likely to emerge: a real object whose radiation is observed with telescopes or a virtual object whose properties are realized by physics-based simulation. Analysis codes marry these two perspectives. Applying common models and methods to observed and simulated populations of galaxies drives forward our understanding of cosmological physics and the astrophysical processes that shape galaxies. Substitute people, financial institutions, or animals for galaxies and the same coupled modes of study apply to social dynamics, economics, or ecology.

The explosive growth in both volume and breadth of disciplinary data has elevated the prominence of data-driven discovery5, one that amplifies the traditional paradigms of experiment/observation and theory/simulation. The challenge, now and in the future, is to make better use of the growing mountains of data within and across the scientific disciplines.

The National Science Foundation envisions an agile, integrated, robust, trustworthy and sustainable CI [cyberinfrastructure] ecosystem that drives new thinking and transformative discoveries in all areas of science and engineering research and education [10]. In a November 2020 planning report, Pioneering the Future Advanced Computing Ecosystem: A Strategic Plan, the National Science and Technology Council calls for a strategic resource spanning government, academia, nonprofits, and industry that will establish an innovative, trusted, verified, usable, and sustainable software and data ecosystem [11] Computational thinking and model thinking are foundational, enabling components of this vision.

---

5We recognize that data has always been critical to science. Here the phrase echoes the Moore Foundation’s $80M Data Driven Discovery research initiative [9], a program funding X-Informatics and Computational-X projects across a variety of domains.
1.2.1 Recent developments within LSA

LSA has recently created a Data Science bachelors degree, discussed below, as well as Quantitative Methods in the Social Sciences (QMSS), a program that “combines data science with social science to prepare students to solve problems, shape policies, and explain a complex world” [12]. The program currently offers a minor degree, totalling 17 credits, consisting of a pair of core courses, a choice among electives, and a senior, team-based capstone project. In May 2021, an initial cohort of five students graduated with a QMSS minor.

The Computational Social Science (CSS) Initiative [13] within the Center for the Study of Complex Systems was originally developed to facilitate research and training in this rapidly growing and interdisciplinary area. The curriculum is a joint effort between a sociologist and a computer scientist. In Winter 2021, it offered an introductory course in CSS. This course was initially part of the QMSS program and will move to Complex Systems in Winter 2022. It uses scaffolded programming notebooks to allow students to run fairly sophisticated text, network, and machine learning analyses without having a huge overhead in terms of programming. A follow up course, currently in development, will train undergrads in the research design and technical skills to independently conduct research in this area.

1.2.2 From Digital to Quantum Computing

While digital computing continues to be realized by von Neumann architectures using silicon chips, a new paradigm of quantum computing is taking shape. The National Science Foundation lists the Next Quantum Revolution as one of its ten Big Ideas [14], and the University of Chicago recently received a $10M expedition grant to support practical quantum computing [15] that involves computer science educators along with physicists and computer engineers. LSA faculty in several natural science departments hold key positions in the Michigan Quantum Science Working Group [16], an umbrella organization for quantum research at Michigan.

1.2.3 Forms of Intelligence and Cognitive Science

Artificial intelligence and machine learning methods will be vitally important to this future. Research in AI spans a diverse range of fields, including computer science, mathematics, statistics, psychology, complex systems, linguistics, and philosophy. The artificial neural networks of AI were intentionally designed to mimic the biological network behavior of the human brain; a set of interconnected neurons fire in a complex manner to produce an outcome in response to a set of inputs. From a functional system perspective, the problem of understanding human intelligence — how cognition and learning happens in biological brains — has close parallels to the intelligence of machines.

Cognitive Science is flourishing in the College, as a research area, a quickly growing undergraduate major, and a graduate certificate. Teaching across the C&D spectrum for all of the College is an opportunity to explore important questions about human and artificial cognition. Programming is a unique cognitive activity which requires (a) representing knowledge in a limited, digital form and (b) prescribing actions for an agent (the computer) that is not human and has a wholly different cognitive mechanism. The field of computing education research is relatively new – its flagship conferences and journals are less than 20 years old. The research questions about how students come to understand and use computing, the relative affordances of different positions on the C&D spectrum, and how to improve both the educational offerings and the technological tools in support of that learning are rich areas for intellectual exploration.

1.3 Computing for Society: Critical Studies of Identity, Expression and Decision-making Algorithms

For most of the world’s population, digital services are the core medium of human interaction and discourse. Writing a note on paper resides in the same realm as playing a vinyl record; both are deliberate rejections of the digital.

Increasingly, ML and AI services are augmenting or replacing humans in decision-making tasks. Robots are not just busy building cars on factory floors, they’re also having conversations with patients in nursing homes to keep their minds occupied. The rationale for the 2019 Michigan Meeting, Living the Digital Life: Objects, Environments, Power, succinctly paints a vision of computing for society.

So pervasive are digital technologies in the 21st century that it is difficult to find critical distance from this immersive new world of ubiquitous connectivity, social media feeds, smartphones, mobile apps, responsive design, algorithmic recommendation systems, and voice-controlled home shopping assistants. While the question “what is the digital?” is compelling, the more pressing question might be instead: what does it mean to be alive in the digital age?

That question certainly resonates with the current Gen Z students of LSA. If there is one safe bet for what their future holds, it is that there will be more machines interacting with them, and more artificial intelligence attempting to guide them. They need to know what’s inside and behind the magical devices they use everyday.
Identity is a fundamental personal asset, and digital identity is central to many pressing concerns about internet-based services. The European Union countries have passed laws giving an individual more control over their digital identity, including the right to be forgotten. But Google, Amazon and other hyper-scale internet companies rely on aggregate collections of data about people to drive their core business models of tailored search and purchasing recommendations. Recent movements such as FATML (fairness, accountability, and transparency in ML) and ethical AI have emerged, and tensions between critics and service providers continue to build. Google recently fired both heads of its internal Ethical AI group, Timnit Gebru in December 2020 and Margaret Mitchell in February 2021 [17], citing policy violations. Many see these firings as reflecting the ethos of a predominantly white male engineering culture, one that spares little time for criticisms of unintended consequences and unfair design while “moving fast and breaking things” in the quest to mint the next batch of tech billionaires.

Algorithms are a form of power. Just as with other technologies, such as airplanes or nuclear engineering, AI and ML methods can be harnessed for good or for ill. There is a need for students in the College to learn more about how machine-based inference works as well as how methods can fail, either in their original intent or via unintended consequences of poor design. The deep learning algorithms enabling facial recognition were not designed to be racist, in fact they’re quite general, but poor execution can lead to racially biased outcomes.

AI ethics and algorithmic fairness are active research topics; both featured prominently in the 2019 MIDAS Symposium [18]. The topic cuts across LSA divisions. Natural scientists employing ML need to understand how a training sample’s construction may bias derived outcomes. Similarly, AI-enhanced loan decisions from a lender, or prison terms from a judge, may perpetuate or even enhance existing social biases. Humanists and humanistic reasoning are essential, both for driving critical analysis and for inspiring productive paths forward.

While Apple, the largest of the hyperscale tech companies, promotes personal security as a core concern, the technologies it employs are personally invasive precisely because the machine needs to identify who its human operator is. An owner of a newer iPhone records a detailed scan of their face for easier access; it’s faster than typing in a six-digit code. Can that user trust Apple to responsibly manage their image, not just now, but forever? Cybersecurity and machine ethics are major societal concerns.

1.3.1 Computing for Expression: New Forms of Media and Property

The computer encapsulates previous media, and affords the creation of new ones. Animated motion pictures existed before Pixar, but few animators forego using computer animation techniques today. The line between live action and animation blur when CGI after effects are added. Computing has changed the music industry, from the experience of the studio to special effects on stage. Distribution of digital media has been changed forever by the Internet.

The whole C&D continuum is touched by the computing for expression theme (see Figure 1). First, digital media can be more easily created, manipulated, and distributed compared to analog forms. Everyone with a computer can use tools like PhotoShop and Gimp for manipulating photographs, GarageBand to create music, and Blender and iMovie for using video special effects that once were beyond the reach of all but the wealthy. But now these tools are also scriptable. With only a few lines of code, a graphic designer can batch process thousands of pictures in PhotoShop. In film studio summer camps, teenagers invent new video effects in Blender that do not yet appear in Hollywood extravaganzas. C&D tools give everyone a creative voice in media.

Video games are a form of expression made possible through computing. The economic impacts of the video game industry are huge – at $90B in 2020, both the music and movie industries are dwarfed in comparison. Video games are also a powerful form of expression. The NYTimes has included games in its online edition which allow users to understand the news through simulation. The phrase “serious games” has been invented to describe the games that exist to teach, to inform, and to influence.

Social media enable us all to be content providers. YouTube, TikTok, and Instagram have created whole new classes of celebrities and enormous wealth for those who master the new media. We now have a new word for these kinds of creators, “influencers.” The influence is significant. Twitter likely played a role in catapulting a reality TV star into the White House. Twitter, Facebook, and other social media have a role in politics, both stable and revolutionary. Future “rigorous and empathetic thinkers, creators, and contributors” will use social media to their beneit, and the leaders among them will invent new forms of computational media or new ways to use the existing.

The notorious rise of non-fungible tokens (NFTs, Figure 2) offers a window into how digital forms have changed art and media. Consider Harry and Charlie Davies-Carr, children whose wholesome but naughty antics – spanning mere seconds – were posted to YouTube by their parents in 2007 and viewed by nearly a billion people across the globe since. In May 2021, a music company in Dubai bought the NFT for this internet meme for three quarters of a million

---

6NFTs are tokens of authentication established with blockchain technology.
Figure 2: Two recent NFTs (non-fungible tokens) illustrate the rising valuations and unconventional range of digital art. At left, Charlie Bit My Finger, a YouTube video viewed more than 880 million times since 2007, was auctioned by the boys’ family in May 2021 for £538,000 ($760,000) [21]. At right, Cryptopunk 7804, sold for the equivalent of $7.6 million [22]. One of a limited series of cyberpunks, that work was created by an AI algorithm and purchased with Ethereum cryptocurrency. Had it been purchased by an internet bot, the lifecycle of creation, certification, and sale of this work could have involved only digital agents.

dollars, money the family intends to use for Harry and Charlie’s college educations as well as carbon offset purchases to compensate for blockchain’s energy consumption [19]. Despite the sale, the boys antics are still watchable on YouTube. The clip was originally to be retired after auction, but the buyer reversed course because “the video is an important part of popular culture and shouldn’t be taken down.” [20]

Dylan Field, the CEO of Figma, hopes Cryptopunk 7804 (right panel of Figure 2) will also be important to popular culture; he wants it to be the “digital Mona Lisa”. The Crytopunks are a series of AI-generated characters produced by two creative technologists (one with a PhD in the biological sciences) who founded Larva Labs. For Mr. Field, number 7804 was special, one of only nine “aliens” in the limited edition of 10,000 punks, and the only one smoking a pipe. He was emotional after the sale, “It was sort of a part of my identity. It was a mask.” [23]

Are NFTs here to stay? Will cryptocurrencies be embraced by the Federal Reserve? Will local news outlets find digital solutions to recapture their influence on public discourse? These are examples of the types of essential questions that LSA students should be prepared to address in their postgraduate careers.

Creating the Digital Studies Institute (DSI) in 2019 was an important initial step for the College in the Computing for Justice and Computing for Expression themes. The institute sees itself as the nation’s first one-stop shop for all things digital, including digital media studies, digital humanities, digital pedagogy, digital art, aesthetic practice and design, and critical thinking about our digital future [24]. Courses associated with DSI, along with QMSS and the Computational Social Science Initiative, also surfaced by our curriculum searches discussed in §2 below. The College also has faculty affiliated with the University’s new Center for Ethics, Society and Computing [25].

1.4 Increasing Demand for, and Limits to, Participation in LSA Computing Education

Computer science and engineering using silicon chips have produced tremendous economies of scale. The same commodity technologies used to power social media sites can, with modest alteration, be rearranged to provide advanced computing services for scientific studies. Networked software and data infrastructure tools enable teams of people to work securely on projects at a global scale.

The LSA Computer Science BS degree, like its Engineering equivalent, is structured to provide students with solid foundations in computer hardware, software, algorithms and theory. As illustrated in the top panel of Figure 3, the size of this program has grown by more than a factor of ten over the past decade. In calendar year 2020 alone, nearly 450 students graduated with a CS degree, and in the College, only Economics has more declared majors. The newer Data Science (DS) bachelors degree, hosted jointly by the Department of Statistics in LSA and EECS in Engineering, is also experiencing rapid growth (bottom panel of Figure 3). A mere five years old, the DS degree is closing in on producing more graduates than the departments of Earth, Physics, and Astronomy & Astrophysics combined.

The Bachelors in CS is the only LSA degree in which no LSA faculty teach. It is the second largest major in the College, but is the most disconnected from the values and culture of LSA. Decisions are made in the LSA BS in CS degree in order to create equivalence with the Engineering BS in CSE degree, not to reflect the growing importance in the College of computing for discovery, expression, and justice.
As of WN2021 term, the fraction of women among students who have declared a Computer Science (CS) degree is 30% \[26\]. This fraction is higher, 40%, for the Data Science (CS) degree, but both lie below the undergraduate population in the College as a whole, which is 56% female. In §2.3 below we present more details of the lack of representation of women and underrepresented minority students in the CS and DS degree programs.

Finally, we note that the number of transfer students within the College has grown from 1840 to nearly 2600 over the past five years, a 40% increase. Accommodating the needs of this growing cohort is a consideration that requires coordination with the community colleges and other institutions that shape the early academic pathways of an increasing fraction of students in the College.

These and other findings discussed below motivate our recommendation to create a richer portfolio of C&D curricular offerings and degree pathways.

1.5 The University as a Leader in Statewide Computing Education

Michigan is one of the least successful states in providing computing education to its school age population. The University of Michigan as the flagship research-intensive higher education institution in the state could play an important role in changing perceptions and raising the priority for CS education in the state.

The Computer Science Teachers Association (CSTA), non-profit Code.org, and the Expanding Computing Education Pathways (ECEP) Alliance released their 2020 report *State of Computer Science Education: Illuminating Disparities* \[27\] which includes a sub-report for every US state. The 2020 report says that 47% of US high schools now offer CS. Michigan (page 56) is at 37%. Michigan is the only state in the US that used to have CS teacher certification and offered pre-service (undergraduate) training in Computer Science, but chose to eliminate it in 2017 for cost-savings \[28\].

In December 2020, the Michigan Department of Education (MDE) released the first *State of Computer Science in Michigan Report* \[29\]. The data collection and writing on the report was led by Aman Yadav and Sarah Gretter of Michigan State with Cheryl Wilson of MDE. A quote from page 11: “The trend of declining course offerings continues at the high school level where even fewer high schools offer CS courses. Code.org course offering data suggests that
only 23.7% of rural high schools, 28% of town high schools, 29.1% of sub-urban high schools, and 21.7% of city high
schools offer CS.” MDE’s numbers are a lot lower than the 37% in the CSTA/Code.org/ECEP report. An explanation
from the authors is that CS is rare enough in Michigan that not everybody who fills out a survey knows what is meant by
“computer science.” A principal might say that they teach CS, when they might mean Microsoft Office or Web design.

Michigan is clearly below national averages on providing CS education to its citizens and creating sustainable CS
education policy. The strategies used in other states may not work in Michigan. Michigan is a local control state, which
means that individual local education agencies (LEAs) can make up their own rules on important issues like CS teacher
certification. In Georgia and South Carolina, the state government has a lot of control in education, so there was a
point of leverage. California is also a local control state, but the California University systems are important to all
high schools, so that’s a point of influence. Massachusetts is again a local control state, but the Tech industry is very
important to the Boston area, and that’s important to the state. Tech isn’t important in the same way in Michigan. The
MDE report shows that there’s a lot of ambivalence about CS in the state. Administrators aren’t that excited about
teaching CS. They don’t see CS education as important for their students. Michigan is a big state, where agriculture
and tourism are two of the most significant industries. Manufacturing is a big deal, but manufacturing workers don’t
necessarily need to know much about computing. Historically, CS has not been viewed as an obvious benefit to much
of Michigan.

This task force argues that computing is important to many interests and disciplines across LSA, but it’s not all
the same computing. Most people think of computing as serving the needs of the Tech sector, those who develop
computing technologies and infrastructures. In LSA, we have faculty and students making new scientific discoveries
with computing, inventing new forms of expression, and critiquing the influence of technology in our society. All of
these require computing education.

The state of Michigan needs more of its students to learn how to build, discover, invent, create, and critique with
computing. LSA alone cannot meet that need. However, the University of Michigan has significant influence in the
state. LSA can take a leadership role in influencing perceptions of computing education. We can tell our story of the
broad uses of computing, the multiple definitions of computing, and the importance of computing education. We can
help our home state help its citizens.

1.6 Computing and the Environment

*We are called to be architects of the future, not its victims.* — R. Buckminster Fuller

Computing with silicon chips uses energy. By 2010, one study estimated the annual consumption of energy by data
centers alone at 1.3% of global, and 2% of US, total electricity usage [30]. Economic and technological forces
have capped data center energy usage at that level up to 2018 [31], but increasing traffic in crytocurrency mining,
blockchain-based transactions, and AI/ML algorithms threatens to raise consumption faster than energy efficiencies can
lower costs.

Building smartphones, the computers in our hands, has substantial environmental impact. A recent comprehensive
study of information and communication technologies (ICT) from McMaster University estimates that ICT accounted
for three percent of global greenhouse gas emissions in 2020 [32]. While more than half of that total comes from data
centers and network equipment, the share from smartphones and notebooks is nearly 20 percent, with much of the cost
associated with the production phase. Along with greenhouse gasses, the ecological costs from mining and processing
the raw materials that make up a modern phone are significant. Modular, long-lasting designs pose a solution. With
support from a European Union Horizon 2020 research and innovation programme, Finland’s Puzzlephone [33] is
developing buzz, but prior projects by giants like Google have largely failed [34].

The University of Michigan, a global leader in sustainability studies, is moving to put that knowledge into practice.
The 2021 President’s Commission on Carbon Neutrality recommends committing to being carbon neutral (inclusive of
offsets) for Scope 1 emissions across all three campuses in 2025, and develop a plan for Scope 2 carbon neutrality by
the same date. Included in the report are strategic recommendations for the curriculum.

- Expand and prioritize carbon neutrality curriculum, training and literacy programs to all members of the U-M
  community across all three campuses.
- Invest in institutional structures to expand and support carbon neutrality focused “living-learning labs” across
  all three U-M campuses.

In partnership with other units on campus (SEAS, Graham, Ford, etc.), the College should keep in mind the opportunities
for leveraging computational and digital studies when responding to this important call.
2 Findings

From November 2020 through April 2021, the TF undertook a number of studies to meet the first of its charges. Courses aligned with TF themes were identified, and a survey sent to 221 instructors of these classes yielded 100 responses, a nearly 50% response rate. Interviews and discussions were conducted with a variety of stakeholders, including LSA Student Government. In this section we surface our findings on the recent history of computing education in the College.

2.1 Summary of Findings

- Views on computing education among LSA faculty vary, with thematic emphasis largely shifting along divisional lines, but support that students in LSA should know about its impact on society — the computing for justice theme — is widespread. Natural science faculty see computing for discovery as important while Humanities faculty lean toward computing for expression. Social science faculty views generally span the two. Our interviews and surveys show that students and faculty in the College feel represented in these themes.

- Across the divisions, more than two-thirds of LSA faculty surveyed agree that LSA should offer more interdisciplinary computing courses jointly with other schools and colleges at Michigan. Opportunities exist to strengthen educational partnerships in support of university priority areas of carbon neutrality, sustainability and anti-racism.

- Many opportunities for computing education exist for LSA students, but organization is poor and a physical space devoted to such support is dispersed. Faculty expressed support for the creation of hubs that help students find resources to learn the computing that they need. Students expressed desires to learn about how computing applications and coding languages can be made more accessible.

- Three introductory courses, one in Statistics and two in Computer Science, enroll more than 1000 LSA students per year. Beyond these, we find that annually just one in five LSA students takes an introductory course, and one in six an advanced course, with significant computational or digital emphasis.

- The LSA curriculum lacks introductory courses to teach students computing and digital technologies that align with thematic and disciplinary contexts. There are no courses in LSA that prepare students to invent with computing, to create novel applications or to blend media like AR/VR with installation art. Educational resources for expression in the College are more heavily weighted on the digital end of the C&D spectrum.

- The core introductory computer science course (EECS 183) is designed to guide students into the rest of the LSA CS major, but only 15% of students who take that course go on to declare the LSA CS major or minor. The programming language used (C++) is poorly matched to LSA needs; not a single LSA faculty member told us that EECS 183 meets their needs.

- Enrollment pressures in EECS courses are so high that Computer Science and Engineering faculty may move to cap enrollment in the LSA CS major.

- The birth sex and ethnicities of current LSA students who declare degrees along the computational-to-digital spectrum vary systematically. Computer Science and Data Science bachelors degree aspirants severely lack women and underrepresented minority students while these groups are well represented in FTVM (Film, Television, Video & Media), Sociology, Communications & Media, Cognitive Science and BCN (Biopsychology, Cognition, and Neuroscience) degrees.

- University-wide institutes in data science (MIDAS), computational discovery and engineering (MICDE), and statistics (CSCAR) support graduate education and graduate certificate programs but currently offer only modest support for undergraduate computational education.

- None of our peer institutions has a broad strategy for meeting the computing education needs of their liberal arts students. There is not yet a US higher-education institution making an effort to ensure that all of their students get the education they need to develop the computational and digital literacy skills required of their future careers as professionals and citizens in an increasingly technological world. LSA has a leadership opportunity.

2.2 Interviews and Surveys

We explicitly sought input from LSA faculty and students tell us what is computing for LSA faculty and students, what do LSA students need to know about computing, and how that computing education should be offered and organized. The first phase involved short essays by TF members.
2.2.1 Early Assessment from TF Member Experiences

As an initial exercise, each of the LSA faculty on the task force wrote an essay on the state of computing education in their own departments. We asked each to assign a grade to the department, which on average was well below the criteria for a Dean’s List. We summarize points distilled these early essays here.

- The most common format for computing education of LSA undergraduates was their statistics or quantitative methods classes. Informal learning is also important, especially for students engaged in independent research projects or courses. These experiences most often support computing for discovery. However, faculty were often not sure what skills and applications were being taught in their students’ statistics courses.
- Only the Digital Studies minor focuses on giving students a critical computing perspective.
- While some departments make extensive use of computing for expression, none of the majors surveyed included a course on how to use computing for expression. There are courses that require students to produce digital artifacts (e.g., podcasts or digital videos), but the artifacts are not a focus of scholarly analysis, construction, or critique.
- The LSA majors surveyed typically do not require students to take advanced courses that feature significant computing.
- Essays mentioned programming languages such as R, Python, and advanced Excel. No one mentioned any classes using C++ (the language used in most CSE courses).

These perspectives, gleaned from a small group of faculty, are limited in terms of scope and utility, but they do reflect the spectrum of interpretations of “computing education”, from primarily computational to primarily digital. The TF then undertook tasks aimed at surfacing attitudes more broadly across the College.

2.2.2 Initial Interviews

We wanted to make sure that we included more diverse perspectives than just from TF members. We used a snowball sampling technique to set up a series of 15-minute interviews. We asked our task force to suggest some LSA faculty and staff who might have perspectives on computing education different from those of our task force members. In those interviews, we asked for more names. Eventually, we interviewed nine LSA faculty and staff (see list in Figure 4). Here are some of the most common themes from those interviews:

Computing as infrastructure. Several participants mentioned how important computing infrastructure is to LSA students and faculty, but how few students know much about it. Digital humanities researchers often create web sites as products of their research, not realizing how different digital artifacts are from paper artifacts. We heard stories about scholarly web sites that became inaccessible because researchers did not understand enough about the ongoing costs of domain names and Internet providers, or because they did not use tools that facilitated preserving and archiving the digital content. Faculty expressed concern that students were so successful at “push button interfaces” that they were not critical or questioning about what was “under the hood.” Students need to understand the technology well enough to see its impacts and be critical of the decisions made when creating it. Several faculty commented that students need to know more about accessibility, that they should be thinking about the digital artifacts they create in terms of who can access them. One participant reminded us that computing for expression is really about critical computing, because both programming and expression are part of cultural practice and that infrastructure also belongs in critical computing because critical infrastructure studies already exist.

Doubts about the impact of computing. Some faculty talked about the pervasiveness of computing, about how it influences all aspects of our lives, and about the importance of understanding and influencing the cultural impacts of computing. Other faculty expressed doubts that computing has had a transformative role in their scholarly discipline. As one participant put it, “Data is not an unremediated reflection of the world,” and humanities scholars already knew that before computers. One stated, “Payoffs with computing have been unremarkable.” Another participant said that we should “start with our learning goals, not with assuming computing.” Most of our participants told us that the most critical learning outcome is for LSA students to be able to reason about program code, algorithms, and technology. One said, “I want deep computing literacy, but teaching coding is a waste of time.”

Diverse goals, including conversational programming and data literacy. Several participants liked “conversational programming” as a goal for some LSA students. Conversational programmers are professionals who have the experience of programming, not to become professional software developers but to be able to communicate effectively with developers, e.g., the manager of a team that includes software engineers. Many of the participants talked about the importance of supporting a range of computing education outcomes for LSA students. One participant told us that she’s “terrified” of CSE courses, and wants LSA computing education to be “more like writing courses, more like the rest
of LSA.” One participant told us that LSA students need to be concerned with “data literacy,” recognizing that data are the result of a set of choices, and the algorithms and policies applied to those data are additional choices. Another participant told us that LSA needs “computing that’s ‘for everybody, not ‘for dummies.’”

Computing for expression is a focus, but LSA is not there yet. Digital humanities scholars told us that much of LSA “isn’t ready for this yet.” They talked about resistance to “born digital” dissertations. Other LSA scholars (those who did not call themselves digital humanists) said that computing as information processing simply “doesn’t fit into my field,” but digital affordances are critical. The ability to express, communicate, share, search, and evaluate with digital manuscripts is a critical skill in many fields in the humanities. One participant told us that the barrier for computing education “is the faculty, to adopt it...To teach it, they have to learn it and see it as integral.” Another participant told us that students need “rhetorical sophistication” to use and apply computing for expression.

2.2.3 Survey of Computing Educators in LSA

We conducted two surveys of LSA faculty about computing education. We asked questions about our three themes and about desirable learning outcomes in computing for LSA undergraduates. We also asked how the curriculum should be designed to support those outcomes. We provided space for open-ended input around computing education.

A first attempt at surveying a randomly selected group of 92 LSA faculty yielded a low response rate; only 14 responses were received. By early 2021, the course search processes had matured to the point where a more targeted set of faculty, those who had taught C&D courses in the past five years, was available. Our second set of invitations went to 221 LSA faculty who taught courses surfaced in the course and syllabus search. From that set, we received 100 responses to our survey, a remarkable 45% response rate, reflecting the fact that those who teach computing in LSA do have opinions to share about it. Based on their self-identifications, the largest share of respondents (43%) were affiliated with the Social Sciences division, 24% with Humanities, and 29% with Natural Sciences.

Our three themes: Computing for Discovery, Expression, and Justice

We asked our respondents, for each of our three themes:

- How closely does this theme describe your use of computing in your scholarship?
- How much does this theme describe what you think LSA undergraduate students should know about computing?

As a preamble, we provided survey respondents with the definitions of the first three themes outlined in §1.1.

Computing for Discovery. Figure 5 presents the results for Discovery. Computing for Discovery is what more than half of the Social Sciences and Natural Sciences respondents reported that they do and relatively few of the Humanities respondents do. Nonetheless, over half of all respondents in all three divisions consider this important for LSA students to know.

Quotes from open-ended comments from faculty highlight the gap between those faculty who value this perspective and those who don’t:

7Survey instrument link.
8Three respondents listed multiple divisions.
“This definition comes the closest of three to my work and what I teach. The data I use come from public sources, require cleaning, sorting, statistical analysis. I also use computing to solve systems of equations, conduct simulations, solve dynamic programming problems, etc.”

“I’m a quantitative anthropologist/biologist and I think that #1 is an outstanding undergraduate major and vital for almost all students to master.”

“My own relationship to computing aligns most closely with ‘model building’ and ‘simulations’ rather than some of the more data-driven definitional components.”

“The definition has no mention of Social Sciences or Humanities so those do not seem relevant. I have never heard of the term fourth paradigm of science before, so I had no reference point for understanding what it meant.”

“No one will know what the ‘fourth paradigm of science’ is. Some examples of ‘complex, often massive, data collections’ would be good.”

Figure 5: Responses to “How much does Computing for Discovery match what you do in your scholarship?” (left); “How much does it describe what you think LSA students should know?” (right)

Computing for Expression. Figure 6 presents the results for Expression. About half of the Humanities respondents recognize this theme in their scholarship, along with over 20% of the Social Sciences respondents. Approximately half of the Humanities respondents believe that this theme is “extremely important” for LSA undergraduate students to know.

Many of our survey respondents wrote very positively about the importance of computing for expression, and described what students should know about this theme:

“Here it is very important to also investigate what gets lost in this form of expression. The tactile, the multi-sensory, the sense of intimacy are hard to convey electronically. This is not to dispute the value of this form of computing, but to emphasize that it comes with tradeoffs, just like any other mode of expression.”

“I try to give all of my students, even those in my substantive courses, a sense of the possibilities for computational expression. This is especially vital in my human rights course where communication and dissemination about human rights ideas is essential for positive change.”

“This is a more general tool than computing for discovery. This is about communicating information, which is core to any discipline and profession.”

One respondent challenged the definition we offered, and was particularly detailed about the educational need for Computing for Expression:

“I think this definition works well for the purposes of emphasizing the role of computation/computers in creative/media arts education. That said, I have a different theoretical framework than is presented in definition #2. While the computer is indeed a meta-tool or a meta-machine, and can simulate any other machine functionally, it is also a creative medium in its own right. The computer has produced multiple new fields of creative arts – as you point out – from interactive arts, networked electronic arts, virtual reality, etc, that are materially distinct from other media. A contemporary and cutting-edge arts education fundamentally includes the computer as an instrument of creative practice. As such, I think it is extremely important for students to learn the foundations of creative coding, digital media arts, and computational methods for creative expression.”
Critical Computing (or Computing for Justice). Figure 7 presents the results for Critical Computing. This theme had a surprising tension between what faculty do and what they want students to learn. Overall, only about 20% of the faculty respondents saw critical computing as “extremely close” to what they do, with more Humanities faculty recognizing this theme in their work than the Social Sciences or Natural Sciences faculty. But there was overwhelming sentiment that this was important for students to know. More than 50% of all respondents rated it as “extremely important” for LSA undergraduates to know.

Several of the respondents contrasted the three themes in their open-ended responses: Discovery, Expression, and Critical/Justice.

“I think this should be a component of their training, but I would not say it is the primary objective of learning computational skills.”

“I think this is critically important, but must be baked into the other two forms of computing.”

“I think everyone (so all LSA students) should know ABOUT this. I’m not sure what depth is required for an informed citizenry, but some level is needed.”

“As a liberal arts school, I think this is where the focus should be. I think some could be borrowed from #1 or #2 on the technical side, but my position is that we don’t need to teach all UG lib arts students how to do X, Y, or Z on computers, but how to understand the processes and the impacts that these can have, especially around the misuse of these systems. I like a hybrid of #2 and #3, with maybe a bit of #1 at a very basic level.”

“I think that all of these definitions are significant, but I was surprised to see #2 & #3 here, and would say that most students and faculty (at least anecdotally?) usually only consider definition #1, even those who (like me) consider definitions #2 & #3 much more connected to scholarship & teaching.”

One faculty respondent argued in favor of adding history as a theme across LSA’s computing education efforts.
“Undergraduate students ought to have more than a passing acquaintance with the history of computing and of information technologies. Perhaps books like this one could be used as texts? [https://lsa.umich.edu/english/people/faculty/hhui/a-prehistory-of-the-cloud.html] Ideally, that instruction would be part of a broader education in the history of science and technology. It’s a shame that the University of Michigan hasn’t got anything like the Collection of Historical Scientific Instruments at Harvard: [https://chsi.harvard.edu/]

**What’s Missing:** In the open-ended questions, we asked faculty respondents about what was missing from these three themes. Some faculty talked about the cultural framing of computing:

“Something along the lines of a reminder that technology doesn’t so much shape culture as culture shapes technology (bringing to the table specific ends and assumptions). Computing is a ‘given’ in the minds of too many students; can one envision other models and implementations of computing besides the one that has grown up in response to our culture’s desires (i.e. consuming, pornography, publicity).”

Several talked about concrete, pragmatic needs in computing.

“Be literate in and capable of critically engaging with complex data applications.”

“Understanding of the methods and algorithms that underly black-box computational techniques. Understanding how to formulate questions that are both substantively relevant and that can be addressed with computational approaches. Understanding of uncertainty, limitations, and biases in computational approaches. Understanding about the data that many computational tools leverage to do their work.”

“Practical applications. Some students don’t even know which way to percentage a column, and it’s not just social science students.”

“Practical applications. Sometimes a model will show A & B are tightly associated with each other. Does that mean A causes B or vice versa.”

“Smaller scale statistical analysis. Critically important.”

An important theme that was emphasized was how different LSA undergraduates would have different needs in computing education.

“My main concern is this assumes all undergraduates need the same thing. I don’t believe that’s at all true. They all need access to all of these things, but only some will need to actually learn or become proficient in them. Phrased as imperatives for all students they frustrate me.”

“Linguistics is a very broad field with intersections both on the humanities and social science sides, so it is very difficult to answer this question broadly for the discipline. In some areas corpus work is very important, and natural language processing skills, but for some sub-discipline areas these skills would be largely irrelevant.”

**Curriculum**

We asked respondents about learning objectives for LSA undergraduates and about how the curriculum should be structured to meet those objectives.

Figure 8 provides the average rating from 1 (“Not at all important”) to 5 (“Critically important (should be required)”) for each objective by the division of the respondent. Social science respondents were most concerned about large-scale data analysis and the ability to build models and simulations. There is broad support for understanding and using tools.

The averages in Figure 8 obscure the strongest messages. Table 1 lists, by division, the percentage of respondents who labeled the given learning objective as 5 – “Critically important, should be required.” For example, though the average of Social Science respondents is high for building computational models and simulations, only 7.1% of respondents thought it was critically important. We have highlighted in yellow those percentages over 10%. Now we see surprisingly strong support for using and picking tools, for communication, and for critique.

**How should we meet these objectives?** We asked respondents to tell us how the LSA curriculum should be structured to meet these requirements.

- There was broad support for each LSA undergraduate program defining how it will achieve computing education goals for its students, with over 50% of the faculty strongly agreeing or agreeing with the proposition
Table 1: Percentage of faculty who rated as 5 (critical, should be required) “How important is it that undergraduate students in your discipline be able to...?” Shading highlights: ≥ 20% support; 10 to 20% support

<table>
<thead>
<tr>
<th>Activity</th>
<th>Humanities</th>
<th>Social Science</th>
<th>Natural Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conduct large scale data analysis</td>
<td>0%</td>
<td>12.5%</td>
<td>25%</td>
</tr>
<tr>
<td>Build computational models and simulations</td>
<td>4.2%</td>
<td>5%</td>
<td>7.1%</td>
</tr>
<tr>
<td>Communicate with scientists &amp; engineers</td>
<td>4.2%</td>
<td>12.5%</td>
<td>14.3%</td>
</tr>
<tr>
<td>Critique science and engineering</td>
<td>12.5%</td>
<td>7.5%</td>
<td>14.3%</td>
</tr>
<tr>
<td>Critique role/impact of computing on society</td>
<td>41.7%</td>
<td>20%</td>
<td>21.4%</td>
</tr>
<tr>
<td>Use computational tools for expression</td>
<td>25%</td>
<td>7.5%</td>
<td>14.3%</td>
</tr>
<tr>
<td>Pick computational tool for purpose</td>
<td>37.5%</td>
<td>32.5%</td>
<td>39.3%</td>
</tr>
<tr>
<td>Critique science and engineering</td>
<td>12.5%</td>
<td>7.5%</td>
<td>14.3%</td>
</tr>
</tbody>
</table>

(Figure 9). Over half of the Natural Sciences faculty agreed that there should be a computing requirement for LSA undergrads, with only about half of the Humanities and Social Sciences faculty agreeing.

- Natural Sciences faculty had the largest percentages of respondents who agreed that faculty should be incentivized to include more computing education in existing classes and to create new computing classes (Figure 10). More Humanities faculty agreed with adding computing education to existing classes than to creating new classes. Less than half of the Social Sciences faculty agreed with either option.

One of the faculty explained their opposition to adding computing content to their courses: “I don’t know how to work this into the above list, but I think there are potential flashpoints between (broadly speaking) more humanistic departments and more STEM departments. That needs to be negotiated carefully. That’s part of why I’m not sure that faculty in general should add computing content to their courses, although I think faculty who can do it should add computing related courses.”

- There was strong support from all three divisions for certificates and other programs to support computing education (Figure 11). The strongest support was expressed for having interdisciplinary courses cross-listed with other units at U-M. The Humanities faculty expressed the strongest support for developing cross-listed programs.

One respondent offered a vision for interdisciplinary courses: “I think it would be useful to have interdisciplinary foundational computing courses that are grouped within similar disciplinary contexts. For example, I would love to see a set of foundational courses specifically for ‘creative coding’. Those could be cross-listed among the ‘creative expression’ departments – such as my department [FTVM] – and Stamps, the School...
of Music, Theater, and Dance, etc. For example, ‘Intro to Programming for the Arts’ or ‘Intro to Creative Coding.’ This would be fantastic.”

![Figure 9: Should each program define its own requirements? Should we make a computing requirement for students?](image1)

![Figure 10: Should faculty be incentivized to add computing education to existing classes? To create new classes?](image2)

![Figure 11: Should there be new programs? More interdisciplinary programs?](image3)

In their comments, faculty suggested that a computing education requirement be a distribution requirement managed by a standing committee:

“If there is a computing requirement it should be like the Race and Ethnicity Requirement, meaning that instructors fill out a form explaining how and why their course meets the requirement. I am not sure that individual departments in the humanities especially know enough about computing to define goals; we might end up with a bunch of digital humanities courses that are what faculty like but completely irrelevant to students.”

“Being in LSA, I think there should be a ‘committee’ (or whatever) at the curricular level that looks for how to bridge the QR, HU and Social Science areas in the context of computing. Perhaps is as
simple as each of these areas of the College compose their own ‘computing curriculum’ or perhaps there should be an over arching cohesive plan. It needs to be discussed.”

“A distribution requirement would help students make time to learn these important skills.”

“Perhaps consider a ‘Computing in the Discipline’ college-wide program similar to ‘writing in the discipline’ style education, or ‘Computing Centers’ similar to Sweetland writing center. Connect these computing in the disciplines courses to career planning & preparation, perhaps in collaboration with LSA opportunity hub?”

“A ‘computing in the disciplines’ requirement (like the old name for the ULWR: Advanced Writing in the Disciplines), with a list of courses developed to satisfy in the same way the ULWR curriculum is developed.”

**Strengths in LSA’s Computing Education**

Several faculty members commented that the current courses in computing education in LSA are working well.

“The economics department has two full computing courses currently and I have been enjoying teaching these courses. The department is motivated to provide students with better skills.”

“I am aware of workshops that introduce basic Python and MATLAB to students. Seems to be working well.”

“My students tend to learn their computing skills from statistics or economics courses.”

“Project based courses that let student choose the platforms and tools that they want in order to get the job done; in my Games course students can use any authoring tool so long as their game works and meets the requirements for the assignment, so some choose ‘easier’ tools like Twine and others use Unity. They are quite good at finding their own support for these tools but it would be even better if LSA-IT would hire staff specifically to assist students, choose tools, etc.”

“Being in statistics, our courses have always considered computing education for our undergraduates. I am not sure about computing education in other programs. In terms of supporting computing education in other programs. In terms of supporting computing education, I think IT support to teaching labs should be as good as IT support to teaching classrooms (which is excellent, one call away). Online support has been good, though.”

DSI was highlighted as playing an important role in computing education within LSA, and respondents stated that it should be grown:

“We already have a computer science department on campus and we don’t need two majors. Adding a certificate or other credential for students, like a minor in C.S. that is attainable within LSA majors, would be a big benefit to our students. C.S. tends to require 100% of students’ attention, to their detriment of the liberal arts mission. Alternatively, if more students knew about The Digital Studies Institute which offers a minor degree that covers all three of the areas you name but especially the last two, this would really meet this need.”

“DSI already provides some of this training within its curriculum and if given lab space to host workshops for students attached to its classes as well as dedicated staff to support class assignments it could really meet this need.”

**Making Change in LSA’s Computing Education**

Several respondents emphasized that computing education in LSA should be faculty-driven, and that we should increase computing education by helping the faculty develop their skills:

“In my view, curricular build-out in this area should be driven by the interests of faculty as well as those of students. Mandating specific forms of knowledge or competence is less compelling than providing and supporting better opportunities for more organic synergies between computationally centered forms of discovery and exploration and research and teaching projects centered in humanities thought and methods.”

“Teach faculty and show them it is not scary. If you don’t want to teach HOW to compute, teach on the critical/analytical side. Additionally, there needs to be an interdisciplinary nature, i.e., STEM students need to learn about the critical parts and HUM students should learn some of the technical parts. Ideally the courses would be interdisciplinary in themselves.”

“In the sciences, computation needs to be a part of every level of the curriculum, and computation should not be forked out to coding classes or the few 400-level computational-based specialized
electives. Sounds great, but from the perspective over in EECS, it is a seemingly impossible needle to thread: the hurdle to re-envisioning classes is extremely high; faculty in science fields themselves lack computational knowledge; and classes tend to focus on historical modes of conducting some science, allowing students at even the 200-level to experience scientific inquiry (albeit using a mode of scientific inquiry 50–100 years in the past)."

Respondents emphasized the goal of helping students to find the existing resources for learning computing on campus.

“I think there are probably a lot of all of these (courses, programs, departments, etc.) What maybe is more productive is to create a ‘hub’ that connects all the disparate courses, modules, programs, certificates, etc. with a coherent vision/mission statement and strategic plan and maybe even a ‘physical space’ and with targeted advising for students. A ‘hub’ also creates an engaged community within the university and beyond — so issues like critical reflection on computing education and community engagement/praxis/outreach through computing education can come to the fore.”

Respondents were concerned about inequitable access to computing hardware and software:

“Computing education can not happen without each student having access to the hardware and software needed to learn the skills required for their field. Ensuring equal access is essential.”

“Not enough lab space. I find that teaching remotely that my class can be larger — not necessarily ideal — and I can teach at any time. The shortage of computer lab space hurts. Students seem to have laptops, but I don’t think they lug them around.”

A strong theme in the respondents’ comments was the need for an LSA perspective in introducing computing.

“I think our goal should be for LSA students to receive more of their computing education within LSA, with a liberal arts/basic science perspective. To me this means not just writing code, but interpreting and critiquing the products of the computation.”

“In my view (for science fields), computing is introduced far too late in the curriculum, and is too heavily siloed into ‘computation’ classes vs. just every other class. There are a few students who (largely self-motivated) do seek out computational exposure at sophomore and junior levels, but for the majority of the few undergrads I see at the 400-level, they are completely new to any aspect of computation, beyond very basic spreadsheet usage. All of our intro to even 300-level labs are built on worksheets or non-computational analyses.”

“I think discipline-based courses are the way to go. Learning requires that the brain sees relevance. Place the skills within discipline-based problems. I can see these being stand-alone courses in some disciplines, ‘labs’ in others, and integrated in still others.”

“I hope that an increasing emphasis on computing will not drive students away from existing disciplines. I think most LSA students should approach computing in context, say within an Anthropology or Chemistry major. Computing removed from any context is mostly just a skill, not a proper undergraduate education.”

“I would love to see (or to build) a set of foundational courses for Creative Coding, Programming for the Arts, and Intro to Digital Media Arts. In my home department — Film, Television and Media — there is currently a lack of courses dealing with these topics, other than my courses (a topic of consensus among faculty in my department). This makes it difficult to get students up to speed on what they should be learning in comparison to students in similar departments at other schools. At my former University, for example, students had a broad repertoire of foundational courses including foundations for Intro to Programming for Arts, Intro to Digital Media Arts (audiovisual), and Intro to Physical Computing in the Arts. This broad foundation allowed for specialization later in particular pathways. I believe that it is essential for all students in a contemporary media production program to have foundations in computational practices in media arts. I would love to see this become a priority at University of Michigan.”

This final quote from a respondent highlights the importance of placing coding/programming in an LSA context:

“For my particular pedagogical needs, it is necessary to have the disciplinary context of media arts when planning foundational computation-based courses. For example, what I find is that I often have to break apart student notions of ‘learning programming’, and that the context of artistic expression is necessary for students who are primarily interested in creative fields. I eschew the notion of ‘learning to code’ when I teach. Code what? What are we making? Why are we making it? What are the ways to make it? Who are the major players in our field? What do they make and how do they do it?”
2.2.4 Discussions with LSA Students

The CETF co-chairs met twice with LSA student government – once with the LSA Student Government Leadership on 18 November 2020, and once with the STAR Subcommittee focused on curriculum on 10 February.

In our first visit with LSA student government, various members told us that they valued computing education because of “marketing, cybersecurity, and ad-tech.” They felt that students with an interest in business and management most needed to learn about computing. As the discussion went on, they saw connections for political science students to understand the role that computing plays in decision-making.

Then the student government leadership started exploring “systemic negative influences.” As we discussed the lack of accessibility in many Web pages, one member argued that there should be “a 10 minute discussion in every course” on how to create accessible digital artifacts. They felt humanities students needed to be reviewing technological impacts in every class. Social studies students should be asking what technology do they really need, and how can technology help in law, medicine, or other careers. They saw a particular value for computation embedded in different domains, like computational legal studies and computational journalism.

The STAR subcommittee was concerned about creating more “hoops” for student in terms of requirements. However, they saw value in students having access to workshops, online courses, or “bootcamps” to help students develop the computational skills they needed for their courses and research. There was significant interest in short courses and workshops, including online/virtual options, over Winter or Spring break. The STAR subcommittee members were also interested in more LSA courses with computing built-in to the structure. Members said that many of their peer students really didn’t see how computing might help them. They see a need for advocates and support for computing education in LSA, because many students “run for the hills from technology.” Some students complained that the existing computing education courses (like STATS 250) was “overly confusing” and “stressful” because the course expected students to use R, but did not explicitly teach fundamentals of coding in R.

The STAR subcommittee members felt most strongly that LSA students should know about critical computing. LSA students should know about the racial bias issues in computing, like poor facial recognition technology. They want students to also be thinking about positive ways to use and think about computing. They explicitly called for more computing for justice minors.

The co-chairs also talked with a couple of LSA students who had tried EECS 183 and considered the LSA CS major, but decided not to declare it. We learned a lot about weaknesses in the current model. Students told us that 183 was fun, but 280 and 281 were much too difficult (“super-intensive”). We heard about “enormous office hour lines,” and the challenges of coding from scratch a program given a 20–35 page specification. Students told us about the mismatch between CSE courses and their other LSA classes, e.g., taking C++ in CSE classes but using MATLAB and Python in Financial Math.

2.2.5 Interviews with Computing Education Informants in LSA

An outcome of our program and survey search and our interviews was a set of specific questions about computing education in the College. To answer these questions, the CETF co-chairs conducted a series of half hour interviews with informants (see Figure 16). In this section, we provide short form answers to our questions, with more details in the Appendix.

What can the existing institutes in Computing for Discovery offer LSA undergraduates?

The two institutes most active around computing for discovery (e.g., data science, computational science) are the Michigan Institute for Data Science (MIDAS) and the Michigan Institute for Computational Discovery and Engineering (MICDE). We spoke with the directors of both of these institutes.

MIDAS and MICDE are both focused on research and community building across units of the University of Michigan. Their missions connect to education at the graduate level. Both institutes include LSA faculty in advisory committee roles and via affiliate faculty status. MIDAS and MICDE might make some of their graduate offerings available to advanced undergraduates. For example, MICDE is interested in exploring undergraduate certificates that connect to their PhD in Scientific Computing. They see a value in supporting a “hub” that could point LSA undergraduate students to resources for computing education, including MIDAS and MICDE resources.

What is currently available for LSA undergraduates around Computing for Discovery?

We spoke with the heads of two programs that serve LSA undergraduate students around computing for discovery, QMSS (Quantitative Methods in Social Science) and Complex Systems. We also spoke to undergraduate leaders in EEB and Statistics. Biology and Statistics do teach at the Computing side of the C&D spectrum, but all groups talked about the struggle to cover all student needs and to staff courses. The biggest problem identified for teaching computing
for discovery in LSA is that access to computing hardware and software is inequitable. We heard arguments for “better communication and coordination” so that students can find what already exists at Michigan.

Why are Communications and FTVM in the center of computing education in LSA?

The curriculum search (§2.3) found that the programs that offered the most courses that met our search criteria for computing-related terms were Communications and Film, Television, and Media (FTVM). We met with leadership in both of those programs.

FTVM is about media production. They use computing for editing (all forms of media) and for construction of media like animation and virtual reality (VR). They have sophisticated server needs in their studios. They don’t currently explore the underlying technical content involved in media manipulation, e.g., how digital video special effects are created, how images and videos are digitally represented, and how media are encoded and decoded (e.g., CODECs). While their classes focus on computing for expression, their faculty take a critical perspective on media. There is a strong ethics theme in all their classes.

The Communications program is about understanding mediated communications, which today, is mostly computational. The most common computing education themes in Communications are for discovery and for justice, with some expression. Communications is significantly about analyzing big data, e.g., with data scraped from Web pages. Communications and SI faculty often do similar things, but SI tends to be more “under the hood” or to enhance the user interface. Communications is about understanding usage pattern to inform stakeholders. Classes in Communication do not teach computing. Specifically, they don’t teach any programming. Communication faculty send their students to the library to learn computational skills.

Both FTVM and Communications leadership want to make computing education in LSA “humanities-based.” Education about coding has to include information about the social repercussions of computing and about the expressive power of digital media and non-linear video editing. They need more faculty who know and can teach computational methods. They also see a value in having a “hub” for brokering and disseminating information on computing education to LSA undergraduates.

Do students in DSI need computing education?

Critical computing is the focus of the Digital Studies Institute (DSI). We had several interviews with DSI faculty exploring if they identify additional computing education needs.

The answer was clearly “yes.” The DSI faculty we spoke with recognized that greater student understanding of computing would enable better informed critiques of computing. For example, a course on the “politics of code” would be much richer if the students had seen and developed (e.g., edited and compiled/executed) their own programs. Faculty told us how progress in their classes was limited because of the uneven experience of computing among students. It’s hard to develop programs when many students have not even seen the HTML that underlies the Web.

However, we also clearly heard that CSE classes as-is are unlikely to meet DSI needs. They would like common, foundational courses to be developed, and DSI faculty would like to be co-developers of any courses in programming for their students. Coding for expression is not the same as coding for application development or discovery. Again, we heard arguments for a “hub” to help students (and faculty and staff) find computing education resources, events, workshops, and courses.

How is the LSA CS major doing?

EECS 183 is the introductory course for the LSA CS major. It has been designed to already address computing for discovery and some computing for expression. There is little about computing for justice in the current course. The goal of 183 is to give students the confidence and skills to go on in the CS major.

CSE is making plans to cap enrollment in EECS 183 and the other introductory courses. Not all LSA students who want CSE classes will be able to get them. This will likely lead to interest in alternative computing options (including degrees and minors) among LSA undergraduate students.

What do organizations that hire LSA majors think about their computing education?

We spoke to a small number of representatives of organizations that hire LSA students. We asked them each about the computing education capabilities of LSA undergraduates. They were surprised to find that LSA students had to be taught how to do basic tasks like managing data in Excel. LSA students did not know how to use scripting or programming for data analysis. Employers told us that LSA students didn’t know what they didn’t know. The employers saw that computing could advance students’ goals, but the students didn’t see how computing could help them.
2.3 Searches of the C&D Curricular Landscape in LSA

Each semester LSA offers thousands of classes across the curriculum. A TF subgroup\textsuperscript{9} undertook searches to surface recent C&D related course offerings with the aim of summarizing the overall scope of current undergraduate course offerings. These findings provide important context for our set of recommendations.

2.3.1 Brief Summary of Curricular Findings

- In the 2019-2020 academic year, three courses (one in introductory statistics and two in programming\textsuperscript{10}) account for nearly one-third of student enrollment in C&D classes. Beyond these three, one in five LSA students enrolled in an introductory C&D course, with introductory cognitive science, econometrics, and multiple physics laboratories hosting high LSA student enrollments.
- At advanced levels, the C&D curriculum reflects the intellectual breadth of LSA and its liberal arts tradition of small classes led by top faculty. Of the 121 advanced C&D courses identified in 2019-20, twenty-six were in the departmental home of computer science (EECS), but social (FTVM, AmCult, Comm, Ling, Soc, etc.) and natural (Math, Stats, Earth, etc.) science departments comprise the overall majority. One in six LSA students took an advanced level C&D course in AY 2019-20.
- Female students along with Black and Hispanic students are significantly underrepresented in the computationally-oriented degree programs of Computer Science and Data Science. In contrast, these groups are overrepresented in the more expressive degree programs of Film, Television, Video & Media, Communications, and Sociology, as well as Cognitive Science and Biopsychology, Cognition & Neuroscience.
- Introductory-level courses tend not to teach computing methods but rather employ them to enhance disciplinary teaching. Unique exceptions in biology and social science\textsuperscript{11} have emerged recently.
- Advanced-level LSA courses do teach computing methods contextualized by subject, most supporting computing for discovery\textsuperscript{12} in the natural sciences.
- The richest C&D course offerings tend to be weakly tied to college requirements; most are electives or special topics.\textsuperscript{13}

2.3.2 Search process

Our reviews use source material available to LSA students: i) course descriptions from the LSA Course Guide (CG), and; ii) syllabi from the LSA syllabus archive. Neither source is complete at the level of the entire University, but the LSA CG contains all courses recognized by the College for internal academic credit and is thus complete in that sense. The syllabus archive is less complete, lacking a good fraction of classes hosted outside the College, especially the College of Engineering.

The syllabus archive does contain past syllabi for a majority of LSA-owned courses. While syllabi can differ dramatically in scope and length, in nearly all cases the information provided paints a much richer picture of the course compared to CG descriptions, which average only a few hundred words. This search mode enables us to surface courses that employ a larger number of C&D terms in their syllabi, a set that we’ll refer to as rich C&D course offerings.

For both sets of materials, we analyzed the frequency of usage of a set of words and short phrases associated with C&D studies, listed in Appendix B. We require multiple matches to minimize extraneous associations, and search team members performed close examination of the results as a further validation step.

The two search modes provide complementary views of the complex curricular landscape within the College. The course description analysis provides a snapshot in the single academic year of Fall 2019 through Summer 2020. For syllabi, we extend the time horizon to the past five academic years, Fall 2015 through Fall 2020, with the goal of exposing a set of rich C&D offerings, including special topics courses or other classes offered sporadically.

Search limitations. Before presenting our findings, we reflect upon the limits of our methodology. Conventions on course numbering vary among the subjects. Course descriptions are dynamic and course titles are often generic

---

\textsuperscript{9} AE, NH, and MRY, with assistance from JoAnn Peraino, Curriculum and Enrollment Manager in the LSA Dean’s Office, and Nick Paulson, Research Assistant at the Institute for Social Research.

\textsuperscript{10} Stats 250 (Intro Stats and Data Analysis), EECS 183 (Elem Programming Concepts), and EECS 280 (Programming and Intro Data Structures).

\textsuperscript{11} Biophys 117 (Intro to Prog in Sci) and QMSS 251 (Comp SocSci).

\textsuperscript{12} Examples include Phys 411 (Comp Phys), Astro 406 (Comp Astro), EEB 480 (Models & Inference), Stats 406 (Comp Data Sci).

\textsuperscript{13} Examples include: Comm 409 (Sem Media Effects), Hon 365 (Cyberscience), AmCult 358 (Topics Digital Studies), Astro 406 (Comp Astro), Phys 411 (Comp Phys), Comm 362 (Digital Media Foundations).
Computing Education Task Force Report

(e.g., Special Topics). Sometimes a single course number operates within it multiple independent courses at the section level (e.g., AMCULT 358). In such cases we have attempted to identify the section-level entities as separate “courses”. Recognizing that any automated search process would be challenged by the College’s environmental complexities, we do not claim to be perfect with respect to completeness or accuracy in our results. Despite the imperfections, our findings help expose the scale and scope of LSA student engagement with C&D education.

<table>
<thead>
<tr>
<th>Subject &amp; Catalog No.</th>
<th>Course Title</th>
<th>2019-20 LSA Enrollment</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATS 250</td>
<td>Introduction to Statistics and Data Analysis</td>
<td>3045</td>
</tr>
<tr>
<td>EECS 183</td>
<td>Elementary Programming Concepts</td>
<td>1486</td>
</tr>
<tr>
<td>EECS 280</td>
<td>Programming and Introductory Data Structures</td>
<td>1141</td>
</tr>
<tr>
<td>EECS 281</td>
<td>Data Structures and Algorithms</td>
<td>723</td>
</tr>
<tr>
<td>COGSCI 200</td>
<td>Introduction to Cognitive Science</td>
<td>598</td>
</tr>
<tr>
<td>ECON 251</td>
<td>Introduction to Statistics and Econometrics II</td>
<td>551</td>
</tr>
<tr>
<td>EECS 376</td>
<td>Foundations of Computer Science</td>
<td>479</td>
</tr>
<tr>
<td>EECS 485</td>
<td>Web Systems</td>
<td>363</td>
</tr>
<tr>
<td>SI 110</td>
<td>Introduction to Information Studies</td>
<td>349</td>
</tr>
<tr>
<td>PHYSICS 141</td>
<td>Elementary Laboratory I</td>
<td>333</td>
</tr>
</tbody>
</table>

Table 2: Top ten courses ranked by LSA student enrollment in the academic year spanning Fall 2019 through Summer 2020 terms.

<table>
<thead>
<tr>
<th>Academic Level</th>
<th># LSA students</th>
<th>% LSA students</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introductory</td>
<td>3637</td>
<td>20</td>
</tr>
<tr>
<td>Advanced</td>
<td>3035</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 3: Reach of C&D courses, exclusive of the Big Three (top three courses in Table 2), for the academic year spanning Fall 2019 through Summer 2020 terms. The table gives the numbers of students enrolled in C&D identified courses at both introductory (100-299) and advanced (300-599) academic levels, and the fraction of the LSA undergraduate student cohort that these numbers represent.

2.3.3 Course Description Search, AY 2019-2020

Based on LSA CG course descriptions for Fall 2019 through Winter 2021 terms, we identified unique classes\(^{14}\) having two or more search items for further inspection. After further inspection of descriptions by TF members, a total of 175 classes are identified, 54 at introductory levels and 121 at advanced levels. We begin by exploring this set of C&D courses.

Scale of Engagement: Enrollment in C&D courses. Table 2 ranks the top ten courses based on LSA student enrollment during the single academic year of 2019-2020 (Fall through Summer terms). Overall, we find that roughly half of LSA students take at least one C&D identified course. The top three courses of Table 2 can be considered foundational in that each enrolls more than one thousand students per year. After this “Big Three”, the largest LSA courses offer education in algorithms, cognitive science, econometrics, information studies, and physics.

Advanced-level courses on CS foundations and web systems also break into the top ten, each course enrolling roughly 400 LSA students per year.

Table 3 summarizes the reach of C&D courses excluding the Big Three. The middle column gives student counts while the final column expresses the numbers as fractions of the entire undergraduate LSA student population. With that reference basis, one in five LSA students enrolled in an introductory level C&D course (beyond the Big Three), and one in six at the advanced level.

These fractions for a single academic year suggest that, over a four year career, an LSA student would, on average, take roughly one C&D course at each level.

\(^{14}\)Cross-lists and meet togethers are counted only once.
Figure 12: Word cloud illustrating the range of introductory level (100-299) subjects represented by a total of 54 C&D courses identified in the 2019-2020 academic year. Font size reflects the number of courses in each subject that have descriptions in the LSA course guide containing two or more C&D search terms. Out of the 54 total, Physics offers seven (six labs and Honors Mechanics), Digital Studies six, and Stats four.

**Introductory level courses.** A more extensive view of the range of C&D course offerings at the introductory level is given by the subjects displayed in the word cloud of Figure 12. A total of 54 introductory C&D courses are identified in 2019-20, and the image scales the font size of each subject proportional to the number of C&D courses offered by the program.

The Physics Department offers six labs and one lecture identified by our matching algorithm. Digital Studies offers six, including DIGITAL 158 (First-year Seminar), 202 (Digital Culture, cross-listed as AmCult 202), and 285 (Social Life in the Digital World, cross-listed as SOC 285).

The description of DIGITAL/SOC 285 poses the following questions to its students:

- To what degree does the digital world reflect and reproduce the culture and structure of the physical society that created it?
- To what degree has the digital world changed the larger society?
- Today’s college students are “natives” in the digital world. As they seek to understand their own identities and the social context in which they exist, will they be helped by the concepts of sociology, most of which were formulated before the invention of computers? Or are dramatic new understandings of social life required?

The description continues, “This course will explore a series of foundational concepts in sociology: structure, culture, inequality, communities, socialization, presentation of self, deviance, institutions, and social movements. In each case, the traditional concept will be explained and its possible application to the digital world will be explored.”

The Department of Film, Television, Video and Media offers three introductory C&D courses, including a Digital Media course (FTVM 202, cross-listed as DIGITAL 220) that critically examines the relationship between new media and society as well as a hands-on survey course in Media Production (FTVM 290). Each enrolls roughly one hundred student per year.

In the natural sciences, the Department of Statistics offers an introductory course in Data Science (STATS 206), an honors introduction to statistics (STATS 280), as well as Making Sense of Data (STATS 150). We find no introductory-level courses in the Department of Mathematics, but this result likely reflects the incompleteness of our simple search methodology. The advanced calculus course sequence (MATH 215 and 216) have lab elements in which students employ numerical methods to visualize and analyze functions of one or more variables.

Economics is represented by four courses, an introductory seminar using the R programming language (ECON 195), an applied data analysis course (ECON 258), and two sections of an introductory econometrics course (ECON 251).

The brand new Quantitative Methods in the Social Sciences (QMSS) program is represented by two methodology courses (QMSS 201 and 251).

---

As a reminder of our methodological limitations, this total of seven should have been eight. The lecture identified (PHYSICS 160) the first of a sequence of honors classes each of which employs numerical simulations as a means to support the learning of physical principles. The second semester course (PHYSICS 260) hints at this feature in its course description while 160 exposes it more explicitly, thereby triggering our search threshold.
Two courses in Applied Liberal Arts (ALA), a program within the College’s Office of Undergraduate Education, deserve mention. ALA 105 (Digital Research: Critical Concepts and Strategies) is a one-credit skill enhancement class focused on information discovery, management, and assessment. ALA 106 (Programs, Information and People, cross-listed as SI 106) reviews “fundamental elements of a modern programming language [python] and how to access data on the internet.” That course is a prerequisite for students seeking to transfer to the School of Information.

Environmental science, the Earth’s climate, and sustainability are represented, as are two courses offered by the Center for the Study of Complex Systems, one on social dynamics and tipping points and another on agent-based modeling.

Computing technologies as framed by science fiction tie into two courses, one offered by the Department of Philosophy, PHIL 154 (Science Fiction and Philosophy), and another from Romance Languages, SPANISH 280 (Conversation Through Spanish/Latin American Film). The course description of the latter explains how it ties to the theme of critical computing, “By means of stories of robots, time travels, dystopias or the perspective of the apocalypse, the course will allow the students to discuss general issues like human relationships or the role of technology in today’s world, and to know more about a wide spectrum of cultural, historical and sociopolitical themes related to contemporary Spain, Latin America and the U.S. Latinx community, such as immigration, war or labor.”

Advanced level courses. The cloud on the left side of Figure 13 shows how subjects are represented in the 121 C&D courses identified at the advanced level, with course numbers in the 300 to 599 range. Again, font size shows the number of courses offered in each subject during the 2019-20 academic year. The image on the right side of Figure 13 shows individual words frequently used in the titles of these courses.

EECS courses are prominent at the advanced level, with a total of twenty six courses. As previously noted, EECS 376 (CS Foundations) and 485 (Web Systems) are large enough to break into the top ten enrollment list of Table 2. Many topics of computational practice are covered by the remainder, including algorithms, advanced object-oriented programming, compilers, database management, embedded systems, machine learning, networks, security, search, and parallel computing. FTVM is home to nine advanced courses that intersect TF themes. Computing for expression is represented by FTVM 391 (Editing Theory and Practice) and the 306/406 sequence (Animation and Digital Media...
Computing Education Task Force Report

Practices I/II) while computing for justice intersects with FTVM 354 (New Media History) and 367 (Digital Media and Identity).

The Mathematics Department also offered nine advanced classes in 2019-20, including three on numerical methods (MATH 371, 471, and 572), one on mathematical models (MATH 462), and one on interest theory (MATH 424) that serve as prerequisites to advanced actuarial courses.

The collection of 121 courses represents the full spectrum of core TF themes. In computing for discovery, ASTRO 406 (Computational Astrophysics) “develops practical working knowledge of the numerical methods most widely used in current research”. Topics covered include “data modeling and parameter fitting, using both maximum likelihood and Bayesian methods”, and application of “various machine learning methods to analyze and classify complex datasets”. Student projects play an important role.

At the intersection of history and digital society, consider one section of AAS 358 (Topics in Black World Studies), the description of which explains, “The course’s [sub-]title, ‘Reconstructing James Baldwin’s Legacy through Black Digital Studies,’ reflects its innovative and experiential focus on the ways in which African American literary history can and should be taught in a rich conversation with the emerging field of Black Digital Studies.”

The right image of Figure 13 illustrates words commonly used in the titles of advanced courses. Nearly one quarter of advanced C&D courses are considered to be topical (27 occurrences of Topics across 121 courses). Many provide an introduction to specific computational methods or systems, such as cryptography or Geographic Information Systems.

Courses focused on data analysis appear multiple times. Table 4 lists examples and shows that the offering from the Statistics Department has a considerably larger enrollment than the others.

<table>
<thead>
<tr>
<th>Subject &amp; Catalog No.</th>
<th>Title</th>
<th>2019-20 Enroll</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANTHRIO/PSYCH 463 /ENVIRON 473</td>
<td>Statistical Modeling and Data Visualization in R</td>
<td>27</td>
</tr>
<tr>
<td>EARTH 468</td>
<td>Data Analysis, Inference, and Estimation</td>
<td>24</td>
</tr>
<tr>
<td>ECON 258</td>
<td>Topics in Data Analysis</td>
<td>65</td>
</tr>
<tr>
<td>DATASCI/STATS 415</td>
<td>Data Mining and Statistical Learning</td>
<td>247</td>
</tr>
</tbody>
</table>

Table 4: Examples of LSA courses focused on data exploration and analysis.

Overall, we find that the advanced C&D course landscape is stocked with a wide range of small, specialized offerings that span the disciplinary breadth of LSA. However, only one in six LSA students engaged with this collection in the 2019-20 academic year. There is room to grow LSA student engagement in advanced C&D courses.

2.3.4 Syllabus Search, Fall 2015 to Winter 2020

While course descriptions provide a valid window into the existing curriculum, their brevity makes it challenging to interpret how deeply the course may engage students in learning C&D concepts, methods or skills.

The search team therefore undertook a separate search involving syllabi with the goal of surfacing rich C&D offerings. The syllabi involved the most recent offering of courses hosted by the LSA Syllabus Archive in terms between Fall 2015 to Winter 2020. As a simple means of automating this search, we define a rich C&D class to be one for which the syllabus contains six or more search terms.

The search yields a total of 137 courses. Figure 14 breaks down this set by catalog level and by division. Three-quarters of the rich C&D courses are advanced-level offerings, and nearly half are in social science subjects.

The number of rich C&D course offerings by subject is shown in Figure 15. Two departments – Communication Studies and Film, Television, Video and Media – are responsible for 35 of the 137 courses identified. COMM courses lean into the critical computing theme, with titles including Political Misinformation (COMM 468), Media and Political Behavior (COMM 329), and Race, Gender and New Media (COMM 424). FTVM offerings touch on both expression and justice themes. Computing for discovery is supported by courses in Math, Statistics, and other natural science departments.

2.3.5 C&D Courses and College Requirements

While the TF did not undertake a comprehensive study of how the courses identified above currently relate to degree and College-level requirements, we note that the Big Three courses often serve as prerequisites that unlock advanced courses across a range of degree programs. Many of the courses satisfy the College’s quantitative reasoning requirement.
Figure 14: Breakdowns of C&D rich course offerings identified through search of AY 2015-2020 syllabi, in terms of academic level and division. Rich offerings are those with syllabi employing six or more unique search terms.

Figure 15: Word clouds showing the number of rich C&D course offerings by subject as determined by search of the LSA syllabus archive. Font size scales with the number of courses in each subject having syllabi in the LSA syllabus archive that contain six or more C&D search terms. Note that cross-listed courses are counted multiple times.

Most courses, especially those identified as rich C&D offerings, serve only as electives to a degree. For example, Computational Physics (PHYS 411) is an advanced elective for the Physics Bachelors of Science degree. Similarly, Computational Astrophysics (ASTRO 406) and Model-based Statistical Inference for Ecology (EEB 480) are electives for the BS degrees in Astronomy & Astrophysics and Ecology & Evolutionary Biology, respectively.

The salience of computational physics is still being recognized by students and advisers; 63 percent of students who graduated with a Physics BS degree in the past ten years took this class, only slightly below the 68 percent who took the second semester advanced laboratory course (PHYS 442) listed as required for the degree.

One of the implementation tasks we recommend is that departments and programs should review their degree requirements in relation to existing and future C&D courses.

2.3.6 The Curious Case of EECS 101

Our search uncovered EECS 101, a course entitled Thriving in a Digital World, that was last offered in WN 2016 term. EECS 101 covered “the fundamentals of computer science and its impact on people, society, and innovation”, with a lecture and laboratory structure. EECS lecturer Jeff Ringenberg developed the course as “a broad introduction to computer science concepts” specifically aimed at non-CS majors.

The textbook for the course, Blown to Bits [35], written in 2008 by Hal Abelson, Ken Leeden, and Harry Lewis based on material from a joint MIT/Harvard course they created, uses non-technical language to explain core technological issues surrounding digital privacy and digital identity in networked society. The book addresses issues that are relevant to the digital lives of all LSA students, and an updated second edition has just been published by Pearson.

The brief trajectory of this course exposes the limitations of relying on faculty outside the College to shape the computing education of LSA students. Instead of thriving and expanding, this course was retired in 2016 as demand for
upper division teaching of EECS courses (Figure 3) grew and Dr. Ringenberg was re-assigned to teach upper level classes to the exploding number of prospective CS majors.

2.3.7 Who is engaged in Computer Science and Data Science degrees?

<table>
<thead>
<tr>
<th>LSA Degree</th>
<th># Students</th>
<th>% Female</th>
<th>% Hispanic</th>
<th>% Black</th>
<th>% 2orMore</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>17,133</td>
<td>56.6</td>
<td>6.7</td>
<td>4.3</td>
<td>4.9</td>
</tr>
<tr>
<td>Comp Science</td>
<td>690</td>
<td>28.9</td>
<td>4.3</td>
<td>1.0</td>
<td>3.1</td>
</tr>
<tr>
<td>Data Science</td>
<td>162</td>
<td>38.3</td>
<td>1.9</td>
<td>1.2</td>
<td>1.2</td>
</tr>
<tr>
<td>Economics</td>
<td>1013</td>
<td>34.1</td>
<td>5.0</td>
<td>1.5</td>
<td>4.6</td>
</tr>
<tr>
<td>FTVM</td>
<td>203</td>
<td>54.7</td>
<td>4.9</td>
<td>10.3</td>
<td>4.4</td>
</tr>
<tr>
<td>Cognitive Sci</td>
<td>230</td>
<td>60.4</td>
<td>3.9</td>
<td>4.3</td>
<td>3.5</td>
</tr>
<tr>
<td>Sociology</td>
<td>252</td>
<td>75.8</td>
<td>10.3</td>
<td>13.5</td>
<td>5.1</td>
</tr>
<tr>
<td>Comm&amp;Media</td>
<td>459</td>
<td>76.6</td>
<td>5.9</td>
<td>4.8</td>
<td>5.0</td>
</tr>
<tr>
<td>BCN</td>
<td>637</td>
<td>78.4</td>
<td>9.6</td>
<td>4.7</td>
<td>4.6</td>
</tr>
</tbody>
</table>

Table 5: Demographic composition of the entire College (first row) is compared to the demographics of students who have declared selected degrees, ordered by increasing fraction of female students, named in the leftmost column. The second column provides the total number of undergraduate students who have declared each degree, with ALL including those not yet declared (roughly half the total population). Subsequent columns list the percentages of students within each degree whose birth sex is female or who declare ethnicities of Hispanic, Black and 2orMore. Relative to the College as a whole, Computer Science and Data Science degrees are similar to Economics (and other natural science degrees, not shown), each lagging in representation by women and underrepresented minorities, especially Black students. Degree programs tied to C&D education that are majority female – Film, Television, Video & Media (FTVM), Cognitive Science, Communication & Media, Sociology, and Biopsychology, Cognition and Neuroscience (BCN) – tend to also engage a larger share of minority students than the College as a whole. (Source: UM Registrar Report service [26].)

We conclude our curricular findings by addressing the question of who is participating in the degree credentials of Computer Science and Data Science.

Table 5 summarizes the demographic composition, in terms of birth sex and ethnicity, of students who have declared these and other degrees in WN 2021 term. The first row summarizes the overall LSA undergraduate population, which is majority female (56.6%) and is comprised of students who identify as Hispanic, Black or Two or More ethnicities at rates between four and seven percent each.

Along with CS and DS degrees on the computational side of C&D education, we include Economics for comparison as it is the largest single degree in LSA and has a strong econometrics component. We also include large degree programs, leaning more toward the digital side, whose courses were identified in our searches for C&D content — FTVM, Communication & Media, and Sociology, and the rapidly-growing majors of Cognitive Science and Biopsychology, Cognition and Neuroscience (BCN).

Moving from more computational toward more digital degrees one sees a growing level of engagement by women. The CS degree is dominated by men, who constitute a 71% fraction, substantially above the LSA average of 43%. The DS degree is also male dominated (63%), similar to Economics 66%. FTVM, while majority female (55%), is slightly below the College average level, while Comm&Media, Sociology, and BCN are dominated by women, with 75% or higher female representation.

The CS and DS degree programs are pathways rarely chosen by underrepresented minority students. Black students, in particular, are scarce. Their one percent share is a factor of four lower than the College average and well below the national average of 5% in CS [36]. The opposite is true for majority female degrees. FTVM and Sociology, in particular, are chosen by Black students at rates that enhance their presence to factors of two to three higher than the LSA average.

These trends are not unique to Michigan. However, a program at Michigan that would expose the common technological foundations of these seemingly disparate C&D communities could offer the means to alert students early in their careers to the value of other degree options, potentially providing more balanced representation by birth sex and ethnicity on both sides of the C&D spectrum.
2.4 Computing Education Initiatives and Peer institutions

The task force conducted two significant reviews of other institutions. In one, we identified novel and interesting programs across the country for providing computing education broadly, beyond computing-specific majors. In another, we investigated peer institutions to identify what they were doing to meet the computing education needs of their LSA-equivalent students.

2.4.1 Novel and Interesting Programs

All Universities are challenged today to provide their students an education that includes computational literacy. We have reviewed some of the more noteworthy programs. Appendix B contains our detailed review.

Curricular structures: If we want students to know something, we usually put it in courses, and maybe incentivize participation in those courses with a degree or certificate. Brandeis has generalized this to a “Digital Literacy” requirement that each program has the responsibility to meet, either through specific courses or through activities in different courses. Wisconsin and University of Texas at Austin offer certificate programs that support blending computing into other disciplines, with less commitment than a minor. Berkeley offers courses related to our Computing for Discovery theme, and USC offers courses and certificates related to our Computing for Expression theme. Perhaps the most novel of these are CS+X certificates or whole interdisciplinary degree programs. Yale has created a whole computational arts program (undergraduate and a graduate degrees), that is a particular slice on our Expression theme.

Both U. Illinois Urbana Champaign and Stanford have created generalized CS+X programs. The University of Illinois has been highly successful (e.g., large enrollments), but Stanford’s collapsed due to a lack of leadership. The challenge of a degree designed to be joint with any discipline and either Computer Science (CS) or Data Science (DS) is to avoid creating a dual degree. Stanford wasn’t able to get the different disciplines to make truly interdisciplinary degrees. UIUC has been able to make the degrees work, but still with few electives, so it’s hard for students to transfer into the CS+X degrees after matriculation.

Organizational structures: Once we decide that students need to know about computing, we have to decide whose responsibility it is. MIT has created a new College of Computing with a novel structure, e.g., their Computer Science department remains in the School of Engineering as well as become part of the College of Computing. University of Wisconsin found it difficult to create a new College, and instead created a School of Computer, Data, and Information Science within their College of Letters & Sciences which combined their CS, Statistics, and Information programs.

Addressing our three themes: Many of the programs we reviewed addressed at least one, and sometimes two of our themes. But no program is explicitly organized to address all three. For example, the MIT College of Computing does not include the MIT Media Lab, an international leader in Computing for Expression.

2.4.2 Comparison with Benchmark Peer Institutions

Using a list of “benchmarking peer” institutions established by the Office of Budget and Planning [37], we investigated how others in LSA-equivalent units were addressing the computing education needs of their students. The summary is NONE of our peer institutions have a comprehensive strategy for meeting the diverse computing education needs of their students.

For each institution, we considered:

- Is there computing education available to all LSA-equivalent students?
- Is there computing education required of all LSA-equivalent students?
- Are there any interesting or novel programs available to provide computing education to non-CS students?

The detailed table appears in the Appendix. While many schools offer some kind of computing course for LSA-equivalent students, most of them are just an option. Two stand outs are Columbia’s Computing in Context course and Harvard’s Using Big Data to Solve Economic and Social Problems. Each of these are applying computing to specific liberal arts topics.

Many schools, like our College, have some kind of “quantitative reasoning” or “Formal Studies” requirement in which computing is an option, e.g., Cornell, Northwestern, Berkeley, UCLA, UIUC, UNC-Chapel Hill, USC, U. Virginia, U. Washington, and Yale. Stanford calls it a “Ways of Thinking/Ways of Doing” requirement. Some of the classes that meet this requirement are interesting for integrating well CS and liberal arts topics. For example, Cornell has a course Introduction to Cultural Analytics: Data, Computation, and Culture which addresses issues like web scraping and data visualization in the context of humanistic research.
However, all of these focus on our theme of computing for discovery. Our peer institutions do not offer courses that meet requirements which we might recognize as computing for expression or justice. We have an opportunity to lead, to distinguish between different uses for computing that fit under an LSA umbrella.

3 Recommendations

We preface our recommendations with the notion that all LSA students need some form of computing education. In 1961, C.P. Snow, author of The Two Cultures, warned about the dangers of an increasingly technological world without a commensurate increase in education about technology [38]. Snow effectively warned that everyone should learn computing in order to understand how computing is influencing their world. He wrote: “A handful of people, having no relation to the will of society, having no communication with the rest of society, will be taking decisions in secret which are going to affect our lives in the deepest sense.” Knowing how this digital world ticks is critical for understanding and contributing to the modern world.

The College has the opportunity to bridge the cultures of the sciences and the humanities using shared foundations in computing technologies. We see computing as a medium rather than as a topic in itself, and see the ability to use computing for discovery, expression, and a search for justice as a kind of literacy.

We challenge the College to realize a vision in which both digital humanists and computational scientists can feel that they are part of a broader community of scholars who seek to push the limits of how computing technologies are designed, employed and evaluated for the good of society and science.

3.1 A Coordinated Vision for Computational and Digital Education

Since at least 2010, undergraduate students entering the University have been born digital, in the sense quoted in the introduction. They are deeply experienced in using apps operating with near-constant access to the global trove, and trashdump, of information available on the internet. Yet, for LSA students who are strongly curious about what powers the digital world, or how to harness its power for good, the Computer Science bachelors degree is still the most visible option within the College, as it has been for the past four decades.

Recommendation #1. Encourage LSA departments and programs to work collectively to galvanize the College’s engagement with computational and digital education by prioritizing interdisciplinary areas of instructional need and by designing new educational pathways and credentials with explicit C&D literacies.

The task force encourages the College to accelerate its investments in computational and digital education beyond the narrow confines of the CS degree. We recommend an implementation plan that pro-actively engages with departments and programs in a curricular inventory exercise aimed at highlighting areas of need and seeding new pathways (courses, degree options, co-curricular experiences) across the core themes outlined early in this report (§1.1). In addition, the College should highlight and expand cross-disciplinary opportunities shared jointly with other units, especially those in which LSA students will engage in C&D work aimed at key global issues.

1a. Foster internal coordination and improve communication. The College should engage a series of meetings coordinating departments, programs, and institutes to meet the C&D education needs of LSA students. While many computing education opportunities already exist, they tend to be disbursed and are often evanescent. Leadership in the College is needed to coordinate and buttress these opportunities and to promote them to students, advisers, and faculty across the LSA disciplines.

We stress that now is the time for each LSA department and program to re-envision computing education for their students. Key questions to be considered include:

• How deeply are computational & digital education integrated into your current undergraduate degree experience? What is working, and what is missing? What aspects could be shared jointly with other departments or programs?
• Does your faculty have the capability and capacity to teach computational & digital literacies in your subject? If not, what is needed?
• Does your department or program currently partner with units outside of the College for undergraduate computational & digital education?
• What steps should be taken to help prepare students to engage meaningfully in computational & digital research or digital media production in your area?

The vision for these activities should be to challenge and enrich the College’s born-digital students with more opportunities to gain knowledge and obtain credentials with clearly articulated computational and digital value. We outline some specific opportunities in our second recommendation and invite the LSA faculty to creatively rise to this occasion.

1b. Highlight and expand interdisciplinary partnerships with other units. LSA graduates are facing a world of increased risks to a sustainable and equitable future. They need to be prepared to contribute meaningfully to problem solving related to epidemiology and public health, human-driven climate change, sustainable resource management, trust in democracy, social justice, improving access to education, and a host of other issues that threaten society and limit scientific progress. Interdisciplinary approaches are critical to making progress on such complex issues, and increasingly these approaches entail data-rich analysis and/or digital campaigns and platforms. The College should work to strengthen partnerships with other units on campus to provide a more diverse set of opportunities for students to acquire skills and experiences that better prepare them to address these areas of global risk.

Of course, LSA graduates do much more than mitigate risks to humanity and the planet. They also endeavor to make our lives worth living by providing meaning, enhancing well-being and enriching the lives of others. Partnerships that embellish LSA students’ abilities in computing for expression are particularly compelling here, as this theme touches the full range of LSA disciplines across the digital-to-computational spectrum.

Regarding the current CS degrees managed by CSE, the committee recommends that the implementation process decide whether to bring the LSA CS major back into the College. This task force has found unique themes for computing education within LSA. With budget constraints in LSA and in the interest of building on partnerships with other units, the LSA CS major might be left with the CSE Division in the College of Engineering. LSA might instead focus on new computing-related degrees related to our themes, which will likely have little overlap with offerings in CSE and SI. On the other hand, the LSA CS major is highly-subscribed, and bringing on LSA faculty who might teach in the CS major might also serve to help with the new LSA-specific computing degrees and courses.

3.2 Curricular Enhancements

While the task force surfaced a number of existing courses having rich computational and digital content, many were ephemeral offerings, developed by a sole faculty member and serving as an elective degree option, at best. A key finding in our review of the curriculum is the lack of introductory courses in the computing for discovery and computing for expression themes. In conjunction with Recommendation #1 above, we invite the LSA faculty to work collectively to address these needs with the goal of establishing a broader and more stable computing education environment for the College.

Recommendation #2. Encourage the creation of new course offerings and the establishment of new LSA credentials, including new majors and co-curricular experiences, across the digital-to-computational spectrum.

We recommend that the College pursue a program of new course and degree program development spanning the three-year implementation period and potentially beyond. This process should be informed by the cultivation step above (§3.1) but can begin concurrent with that step in the first year of implementation.

There is some degree of urgency based on the CSE Division plans to reduce the number of seats in CS classes available to LSA majors as early as Fall 2021. We propose that LSA aim to have new computing degree options available to promote to LSA students by Fall 2022.

2a. Offer new introductory courses in computing for expression and for discovery. The default introductory computing course (EECS 183) does not currently meet the needs of LSA students broadly and its recent enrollment growth is simply unsustainable. Creating alternative introductory courses that align with LSA themes of justice, expression, and discovery is an imperative step that would resolve a pressing need identified in our findings. Whether these classes should count as equivalent to EECS 183 in the LSA CS major is an open question for the implementation process to resolve. Potentially the follow-on courses (EECS 280 and 281) could be modified so that they did not presume prior experience in C++, so that introductory courses in Python, R, or Processing (for example) might serve as adequate pre-requisites.

2b. Encourage additional curricular and co-curricular opportunities. Faculty support for interdisciplinary computing courses that are cross-listed with other schools and colleges is high (Figure 11) as is support for certificates or other credentials. LSA should review existing opportunities, including short courses offered by CSCAR, MIDAS and MICDE, and identify options that support the educational needs of the three core themes. The Opportunity Hub could
work to increase the number of internship opportunities that explicitly employ a range of digital and computational skills. The LSA Development Office could seek donor funding to establish summer fellowship and internship programs in which students use their computational and digital skills to address issues of global importance. These opportunities could be offered in partnership with other units, such as SEAS, SI, and the Graham Sustainability Institute.

2c. Take steps to grow DSI and QMSS/CSS into major degree options. The Digital Studies Institute has led the effort to provide a broad liberal arts perspective to digital education with respect to the themes of computing for expression and for justice. The new Quantitative Methods in Social Sciences program aims to support education in the computing for discovery theme as applied to the social sciences. The courses produced out the Computational Social Science Initiative are also highly relevant to that theme, and were developed in a close collaboration between a sociologist and a computer scientist. Both DSI and QMSS currently offer minor degrees. Perhaps QMSS and Complex Systems (which houses the Computational Social Science Initiative) could combine their offerings to create a contextualized social science major where students would work with authentic, real-world data and problems. These efforts should be expanded to become viable bachelors degree options for the College’s born-digital student body.

2d. Create more mixed degree options, such as DS+X and integrated CS+X degrees. Encourage departments to engage with the Department of Statistics to broaden the data science degree into DS+X options that seat data science methodologies within traditional domains. In our search of innovative computing education programs (Section 2.4), we contrasted Stanford’s failed CS+X degree and the relative success of the Illinois CS+X degrees. The 17 CS+X degrees at Illinois have a combined enrollment larger than that of their CS degree. A flaw of both programs are that the CS and X components are largely disparate. At Stanford, the CS+X degree was essentially a double major. At Illinois, it’s a bit better, but these degrees still have few electives and are difficult to transfer into. We hope that Michigan can do better and have truly integrated degrees, that combine computing with another program in LSA in such a way that students who discover an interest in computing after their first year might still be able to complete the degree on-time.

2e. Support transfer student pathways with a threaded curriculum. The current LSA CS major is unfriendly to transfer students. It’s very difficult for students transferring into Michigan to get much credit in the CSE program. LSA is taking on an increasing number of transfer students. Pathways need to be created that enable students transferring in to progress towards their degree program without losing time and credits.

One way in which we might create degrees that are truly integrated between CS or DS with other LSA programs is to created “Threaded” curricula. LSA might define threads of Computing which could be combined with threads offered from other LSA programs. A threaded degree program can offer students more flexibility than a CS+X or DS+X degree program. At Georgia Tech, computing is defined in terms of eight areas (such as “Devices,” “Information Internetworks,” “Media,” and “People”). Each of these areas has a set of courses associated with them, to prepare a student to work in that area of computing. The definition of the BS in Computer Science at Georgia Tech is to complete any two threads [39]. They have now generalized this model.

- The BS in Computational Media which is joint between Georgia Tech’s College of Computing, Ivan Allen College of the Liberal Arts, and the School of Music is defined as combining any two threads from Computing (Media, People, or Intelligence), Liberal Arts (Film & Media Studies, Game, Interaction Design), or Music Technology [40].

- The BS in Computer Engineering is joint between Computing and Engineering It consists of the three computer engineering foundation threads (Distributed Systems & Software Design, Cybersecurity, Computing Hardware & Emerging Architectures); three Computer Science threads (Devices, Information Internetworks, Systems & Architecture); and three electrical engineering threads (Robotics & Autonomous Systems, Telecommunications, Signal & Information Processing). Students satisfy degree requirements by selecting two of the threads with at least one thread being chosen from one of the three computer engineering foundation threads.

Growing the C&D curriculum in the College will require imagination, will, and resources. Developing these new courses and programs will require course development assistants, relief from existing teaching obligations, and assistance from the experts who help implement the best teaching practices for computing education and maximize undergraduate student engagement. The implementation team will need to develop visionary examples of what C&D means for the College. The task force recommends the definition of personas, descriptions of students with passion and goals whose success would be enabled by C&D education from the College. The task force sees C&D education as critical for enabling students, and that vision and potential needs to be communicated across the College.

3.3 A Home for Computational and Digital Education in the College

Computational and digital education needs to be sustained within LSA. The initiatives of individual faculty might start computing education in LSA, but it needs an academic home if we want it to be sustained. The existing computing
education opportunities within LSA (e.g., in Communications, Statistics, and Economics) meet particular needs. Our findings suggest that computing education in LSA will need to be heterogeneous – there are different kinds of needs that we have clustered into themes. However, all students need some kind of computing education. A strategic and comprehensive effort should be housed in a program or department with the explicit charge to meet the computing education needs of the entire LSA undergraduate student body.

**Recommendation #3.** Embrace ownership of computing education within the College by designating a sustainable home for coordination and future growth.

We outline a pair of options to achieve this end. The eventual solution may involve either or a mix of both.

**3a. Option 1:** Create a new program, tentatively referred to as the Program in Computing for Society and Science (PiCSS). PiCSS could be a lightweight, umbrella organization that serves as an information hub and access point for C&D opportunities in the College. Or, it could be a program that is housed in a more ambitious Institute for Computing for Society and Science that would combine responsibility for computing education (through PiCSS) with an interdisciplinary scholarly and scientific mission of advancing knowledge and practice in all three of the foundational themes.

We see PiCSS as having three main roles.

- **Be the “hub” for computing education in the College.** There are courses, workshops, and other learning resources for students to use for their computational and digital education. These are spread across departments, programs, and institutes. PiCSS could provide advisors and maintain Web resources to help students meet their programmatic and personal computing education needs. PiCSS would coordinate closely with DSI, QMSS, Complex Systems, Statistics and other LSA departments, and liaison with external units such as EECS, MICDE, MIDAS, and CSCAR to drive progress on curricular changes in Recommendation #2. Programs might be able to define their computing education needs in terms of outcomes, and advisors in the “hub” could help match students to the resources that have been identified as helping students achieve those learning outcomes.

- **Offer the foundational courses for all three themes.** A foundational course would teach the necessary computing concepts and skills for further study in a particular theme, which may include programming. There are some foundational courses for our discovery theme scattered across departments, but they may not meet all students’ needs. We have identified no foundational courses for our expression or justice themes. PiCSS could offer these.

- **Provide collaborative faculty for C&D offerings across the College.** We found that students and faculty are interested in computing education integrated across course and program offerings, not just in a set of foundational courses. However, many faculty mentioned that they need support because they do not have the necessary C&D skills. PiCSS faculty (which may be partial appointments shared with other programs and may include lecturers) would be experts in computing education and could collaborate to team-develop and team-teach courses that bring in C&D elements across LSA.

**3b. Option 2:** Expand the current program in Cognitive Science by creating a new Department of Cognitive and Computational Sciences (CCS) charged with comprehensive oversight of computing education. This step would represent a half-century long full circle back to the LSA origins of the CS degree (§1). A CCS department could serve the same roles as PiCSS with an important additional advantage. In one model that is a blend of the two options, PiCSS could exist as a distinct comprehensive curricular program, but whose primary academic home is a CCS department. A department comes with an important advantage: If we want tenure-track, research-intensive faculty to be involved in these activities, we need to also give them the opportunity to work with graduate students. We can identify faculty (existing or new hires) who would be interested in meeting the computing education needs of LSA students and who are interested in the research questions associated with that activity described earlier, many of which would fall naturally within the scope of a cognitive science department grounded in computation.

The recommendation to develop these two options would be in addition to the recommendation to develop the DSI and QMSS majors. Those majors would have a home. Foundational courses and faculty would need a home, as well as the “hub.” Either of the above program options could serve as host or co-host of the DSI and QMSS majors.

**3.4 Dean’s Office Leadership**

The current structure of the LSA Dean’s Office reflects a perspective on computing as a set of services to be managed; the highest position (Chief Information Officer) is a staff position reporting to the Chief Financial Officer. There is no
faculty voice to provide the Dean and Associate Deans with an intellectual vision and faculty-informed guidance on the
digital and computational functions of the College.

**Recommendation #4.** Establish an academic leadership position for a tenured faculty member to
manage Computing, Information and Data for the LSA Dean’s Office.

While some may see this recommendation as lying beyond our original charge, the TF members feel that this recommen-
dation is a natural outgrowth of the previous three. Cultivating a shared culture across the entire digital-to-computational
space will require leadership, and the implementation phase especially will benefit by high-level coordination within
the Dean’s Office. This position could be permanent or exist for a period of time, at least five years, with the primary
aim of launching the curricular initiatives outlined above.

The high end version of this recommendation would be an Associate Dean for Computing, Information and Data
(AD-CID). The faculty member in this role could critically monitor the health of the C&D curriculum, coordinate
policies on identity and data management, assist in the design and maintenance of the IT service portfolio, and manage
projects aimed at better utilizing existing LSA IT resources in service of the College’s research, teaching and public
service mission.

The person could also serve external roles, such as representing LSA in the University’s Information Technology
Council. Considering the fact that a number of LSA faculty have partnered with UM’s Center for Academic Innovation
to develop new instructional services that are widely used by students and faculty in the College (including Atlas,
E-Coach, GradeCraft, M-Write, Problem Roulette, and Viewpoint) the AD-CID role could assist in curating these
services and promoting new innovation by LSA faculty. The AD-CID could also oversee the presence of LSA in
Michigan Online.

We can see a cost for *not* having such leadership. Stanford’s CS+X degree program failed, in part, because there was no
Dean-level leadership around the program. The program was a collaboration between CS and any other programs that
wished to define a joint degree. There was no leadership bringing programs to the table to negotiate, and no one to
encourage the programs to compromise to free up more electives for students. Illinois has had that level of leadership,
which is why their CS+X programs have been successful.

**Acknowledgments**

The Task Force is indebted to Traci Trimmer for logistical support, JoAnn Peraino and Nick Paulson for supporting
critical analysis of the recent curriculum, members of LSA Student Government who invited TF members to their
zoom meetings, and members of the LSA Advisory Team: Benjamin Anderson, Director of Employer Development,
LSA Opportunity Hub, JoAnn Peraino, Curriculum and Enrollment Manager, Michael Hartman, Assistant Director,
Transfer Initiatives and Partnerships, Shylynn Lofton, Comprehensive Studies Program (CSP) Faculty, Tyrone Stewart,
Academic Advisor, Academic Standards Board, and Tim McKay, Associate Dean of Undergraduate Education.

Finally, we thank the one hundred members of the LSA faculty who took the time to participate in our survey of
computing educators as well as all the many other members of the University of Michigan faculty and staff who agreed
to provide input via virtual meetings necessitated by the Covid-19 pandemic.
Appendices

A Charge for an LSA Computing Education Task Force

Timothy McKay, Associate Dean for Undergraduate Education
August 2020

Background and motivation:
Research in computer science at Michigan began in 1949, with a project led by Assistant Professor of Philosophy Arthur Brooks. After the UM Statistical Research Lab acquired our first on-campus computer, new Assistant Professor of Mathematics Bernard Galler taught LSA’s first course in programming - Math 73 - in 1956. Burks and Speech Professor Gordon Peterson launched a graduate program in Communication Science the following year. Their first PhD student, John Holland, graduated in 1959. Holland went on to become Professor of Psychology, Complex Systems, and Computer Science at Michigan, and an icon of interdisciplinary science inspired and enabled by computation.

In 1961, LSA expanded the Communication Science program to add an undergraduate degree, and in 1964 a Department of Communication Sciences was launched. It became Computer and Communication Sciences (CCS) in 1968. It was not until 1984 that the LSA CCS program was brought together with COE graduate programs to create the Electrical Engineering and Computer Science Department. While EECS has ever since been fully responsible for the CS program, the degree remained an LSA one. This unique arrangement creates significant tensions. The College of Engineering seeks to train computer science professionals, while LSA students seek to develop computational expertise as part of a broad liberal arts education.

Today, computing is ubiquitous in disciplinary practice and society at large. Far more people use programming as a tool in their professions than practice programming as the focus of their profession. LSA graduates will put computers to work in their careers and their lives. We have a responsibility to prepare all LSA students to use these new computational literacies at the highest levels. We want them prepared to magnify their voices and lead with a reflective practice that recognizes both the power and the perils of computational technologies.

Launching an LSA Computing Education Task Force

To help the University of Michigan to meet this responsibility, the LSA Dean’s office will create an ad hoc LSA Computing Education Task Force (CETF), charged with imagining a program that could prepare all LSA students to gain and leverage computational literacy. This preparation should take place at multiple levels. All students should achieve some basic competency; others should have opportunities to focus on these topics at the level of academic minors and majors. This work must be situated within the context of other computational learning opportunities taking place at Michigan, especially in the College of Engineering and the School of Information.

The Task Force is charged with three primary goals:

1. Explore what computing is being taught to LSA students, and to whom, and identify unmet needs. Computing already plays an important role in many LSA classes, but there has not been a college-wide effort to (a) identify the computing knowledge and skills needed by LSA students and (b) explore whether and how we are meeting those needs through learning experiences in existing classes. The task force is charged with surveying LSA to identify what is currently being taught about computing, who is participating in those learning opportunities, and what additional elements would most effectively advance LSA students’ computational literacy.

2. Develop plans to provide computing education for LSA students. Meeting students’ needs and considering their feedback for computing education will likely involve adding new learning opportunities to existing classes, creating entirely new classes, considering a distribution requirement, new majors, even new departments or programs. It will also involve collaboration with other schools and colleges. The task force is charged with proposing plans to ensure that future LSA graduates will be able to use computation to better understand the world, engage effectively with society, and lead responsibly. Their work should include input from LSA students, faculty, staff, and alumni. This input may come from advisory groups, surveys, town hall meetings, or other mechanisms that the CETF finds useful.

3. Draft recommendations for implementation over a three year period. The task force should draft recommendations for implementing this plan over a three year period. These preliminary plans will be discussed by the LSA Executive Committee and used as a starting point for further development by the Associate Deans of the College, in collaboration with the Curriculum Committee and relevant LSA Departments and Programs.

---

16The charge was later revised to drop this goal.
Task force membership: This task force will include nine faculty members, with six from the College of LSA, two from the College of Engineering, and one from the School of Information.

Task force timetable and support: As this task force is being launched during a global pandemic, we are particularly cognizant of costs and pace. We are all being asked to do more and different things than we have in the past. At the same time, the pandemic makes us even more aware of how critical computing and communications technologies are to our work, and our physical and mental health. We must prepare LSA students to use computation effectively in all aspects of their lives, and we cannot delay this effort.

Task force members will be recruited in August 2020. The group will meet bi-weekly during the fall 2020 and winter 2021 semesters, delivering a report to the LSA Executive Committee by May 1, 2021. Associate Dean Tim McKay and the Undergraduate Education Advisory Group will meet with the Task Force three times a term to provide input and updates. Staff support for the Task Force will be provided by a part-time graduate assistant supported by the LSA Division of Undergraduate Education.

Each Task Force member will receive a one-time $2000 research fund to support their participation.
B Searches of Course Descriptions and Syllabi: Digital and Computational Terms

The search team began with the following terms originally used to search for courses related to the DSI minor:

- digital
- technology
- internet
- social media
- Instagram, Twitter, Facebook
- fake news
- surveillance
- new media
- trolling
- bots/robots
- video games, gaming
- virtual reality, augmented reality
- algorithm
- computers, computing
- artificial intelligence
- 21st century
- posthuman

The TF then added a set of terms leaning toward the computational side of the C&D spectrum:

- computation, computational
- software, programming, source code
- simulation
- visualization
- machine learning
- object-oriented
- agent-based model
- extended reality
- quality assurance, verification, validation
- broadband, network, bandwidth, ethernet, cyberspace
- data, dataset, byte, binary
- information systems, platform
- authentication, authorization
- open source, Unix, Red Hat
- cloud computing, virtualization, virtual machine
- application program interface
- microprocessor, mainframe, supercomputer
- hyperlink, hypermedia, hypertext, multimedia, usenet
- chipset, compiler, machine language
- c++/c#, python, java, javascript, visual basic, PHP, SQL, ruby, matlab, julia, jupyter, swift, assembler (Note: c and R were omitted)
- API, ASCII, DBMS, DHCP, FTP, TCP/IP, VMS, GUI, IRC, CPU, RAM, ROM, RAID

40
References

[40] Georgia Tech – Bachelor of Science in Computational Media, Jul 2021.