✅ Major Worksheet

DCS Major Requirements Guide - Classes of '27, '28, ‘29

We recommend declaring DCS no later than Feb 1 of your Junior year.

https://www.bates.edu/digital-computational-studies/major/

The DCS major will consist of **11 courses**, covering fundamental aspects of computer science, data science, and critical digital studies. Students will learn to build, analyze, and critique digital solutions while engaging in interdisciplinary collaboration.

	Requirement	**Possible Options (circle choice)**	**Semester**
CS Methods	Intro to Computer Science	one of: DCS 109{C,D,R,S,T}, DCS 111	
	Software Dev & Synthesis	one of: DCS 211*, 229	
Core Methods+	DCS: Data Science & Analysis	e.g.: DCS 105, 117, 204, 211, 212, 307, 375	
	DCS: Critical Digital St.	e.g.: DCS 105, 106, 204, 206, 212, 236, 301C, 305, 347	
	DCS: Human-Centered Design	e.g.: DCS 106, 170, 236, 301C, 304, 305, 325 S31, S33	
	DCS: Community Engagement**	e.g.: DCS 106, 109C, 170, 236, 301C, 304, 325	
	Two 300-level or S40-level DCS Electives++		
	Senior Capstone	DCS 456 (meets W3 requirement)	
	Additional Electives to reach 11 total courses***		

* Students who complete both DCS 211 and 229 may apply 211 to the DCS: Data Science & Analysis core method (yellow).

+ At least two distinct courses are required to complete additional core methods (yellow).

** An approved CEL-tagged course outside DCS can be substituted.

++ Not including DCS 360, 456, 457, or 458. Students may use one Short Term numbered S40 or higher toward this requirement, not including S50 *Independent Study*.

*** Up to **two** Short Term courses (s20 or higher) can count towards the major, and up to **three electives** may be taken from institutions external to Bates (e.g., study abroad, Roux) with prior approval by the DCS Program Committee. One course can be from outside the major - this can be your CEL course or, with permission, another elective related to DCS, but not formally cross-listed with DCS.

In addition to these requirements, all Bates graduates are required to complete the general education requirements for the catalog in which they have graduated.

Students who declare the DCS major cannot declare the DCS minor nor DCS GEC (C093).

It is the responsibility of the student to verify whether their Bates "plus-one" has any restrictions on double dipping requirements across majors or minors.

❓ About the major

Key Program Requirements for the DCS major

Learning Outcomes for the Major

Graduates of the DCS major will:

- Develop strong **computational and data analysis skills**.
- Apply **teamwork and communication** in digital projects.
- Critically assess the **social and ethical impacts** of technology.
- Design and implement **innovative digital solutions**.

The DCS major will consist of **11 courses**, covering fundamental aspects of computer science, data science, and critical digital studies. Students will learn to build, analyze, and critique digital solutions while engaging in interdisciplinary collaboration.

Core Elements (5 Courses Minimum)

1. **Computer Science Methods (2 Courses)**
    - **Introductory Course:** DCS 109C, 109D, 109R, 109S, or 109T (Fundamentals of Computer Science & Computing in Society) Note: current students can sub 109/111.
    - **Software Development:** DCS 211 or DCS 229 (Advanced Programming & Algorithmic Thinking)
2. **Methods Courses (4 categories, with no course fulfilling more than two categories, so this is 2 - 4 courses)**
    - **Data Science & Analysis:** DCS 105, 204, 212, or 307
    - **Critical Digital Studies:** DCS 105, 106, 204, 206, 212, 236, 301C, or 305
    - **Human-Centered Design:** DCS 106, 236, 301C, 304, 305, 316, 325, or S31
    - **Community-Engaged Learning:** DCS 106, 109C, 236, 301C, 304, 316, 325, or an approved CEL-tagged course outside DCS
3. **Capstone Course (1 Course)**
    - A **senior-year** project-based course where students work in teams to design, develop, and critique digital solutions in partnership with external stakeholders

Electives (Up to 5 Courses)

- At least **two** electives must be **300-level DCS courses** (at least 1 before capstone).
- Up to **one** elective may be an approved non-DCS course.
- Up to **three** electives may be taken externally (with prior approval).

- Up to **two** Short Term courses (s20 or higher) can count.

Key Definitions

Bucket Criteria Definitions

- **Data Science & Analysis:** Courses focusing on data collection, wrangling, analysis, visualization, and statistical methods.
- **Critical Digital Studies:** Courses that explore social, ethical, and historical implications of computing and digital technologies.
- **Human-Centered Design:** Courses emphasizing design thinking, usability, accessibility, and user experience.
- **Community-Engaged Learning:** Courses incorporating partnerships with external stakeholders for real-world impact.

DCS 4XX: Design Capstone

- A culminating experience where students work in **teams** to design, develop, analyze, and critique a public-facing digital solution.
- Focus areas include **computing ethics, human-centered design, data analysis, and stakeholder collaboration**.
- This W3 course involves **storyboards, prototypes, oral presentations, software artifacts, and documentation** with regular feedback from stakeholders.
- Build your ePortfolio for employers.
- Students may apply for a thesis as well, but must complete capstone regardless.

Will this be a STEM OPT approved program?

We propose classifying the major as "11.0101 Computer and Information Science, General". The description for this code is:

> *"A general program that focuses on computing, computer science, and information science and systems as part of a broad and/or interdisciplinary program. Such programs are undifferentiated as to title and content and are not to be confused with specific programs in computer science, information science, or related support services."*

Note that this code is part of the STEM Optional Practical Training (OPT) program.

📋 Courses

I. Descriptions of Existing Courses in DCS

DCS 105 Calling Bull: Data Literacy and Information Science
Our world is rife with misinformation. This course is designed to hone digital citizenship skills. It is about "calling bullshit": spotting, dissecting, and publicly refuting false claims and inferences based on quantitative, statistical, and computational analysis of data. Students explore case studies in policy and science and dissect the "who, what, where, when, why, and how" of bullshit propagation. Examples include election misinformation, interpreting health risk, facial recognition algorithms, and science communication. Students practice visualizing data; interpreting scientific claims; and spotting misinformation, fake news, causal fallacies, and statistical traps. In doing so, the course offers an introduction to programming with R for data analysis and visualization. MOI: [QF, SR]

DCS/GSS 106 TechnoGenderCulture
Two premises inform this course: technologies have histories and cultures; technologies are gendered. The course brings together the disciplinary approaches of science and technology studies and gender and sexuality studies to explore contemporary problems at the intersection of gender and technology. Students explore classic texts in these fields and undertake design processes that help them apply those texts to real-world problems. MOI: [AC]

DCS 109D Intro to Computer Science for Data Analysis
This course is an introduction to computational thinking and problem solving via programming, designed for students interested in addressing problems in brain research, as well as experimental science more broadly. Students learn fundamentals of computer programming using Python, including basic data structures, flow control structures, functions, recursion, elementary object-oriented programming, and file I/O, as well as discussion of higher-level concepts including abstraction, modularity, reuse, testing, and debugging. By implementing programs in contexts such as image processing, neural networks, and the analysis of electrical brain activity, students develop an understanding of computational problem solving and gain experience in broadly applicable software development for data analysis. MOI: [QF]

DCS 109R Intro to Computer Science for Robots
This course introduces computer science, computational thinking, and problem-solving in the context of robots. Students learn about computing in terms of the representation and manipulation of data, fundamental algorithms, and societal implications of computing. They will learn the fundamentals of computer programming using Python, including conditional statements, iteration, abstraction, testing, modularity, and debugging. Students will gain an understanding of computational problem solving through implementing programs to control robots and solve robotics problems. MOI: [QF]

DCS 109S (formerly 109) Intro to Computer Science for Software Development
This course is an introduction to computational thinking and problem solving via an introduction to computer programming, designed for students interested in broadly

applying computing and software solutions across a range of disciplines. It considers computing as a discipline of study, exploring the representation and manipulation of data, fundamental algorithms, efficiency, and limits of computing. Students learn fundamentals of computer programming using Python, including basic data structures, flow control structures, functions, recursion, elementary object-oriented programming, and file I/O, as well as discussion of higher-level concepts including abstraction, modularity, reuse, testing, and debugging. By implementing programs in contexts such as image processing, voting algorithms, DNA sequence analysis, and simple games, students develop an understanding of computational problem solving and gain experience in broadly applicable software development skills. MOI: [QF]

DCS 109T (formerly 111) Intro to Computer Science for Text Analysis
This course is an introduction to computational thinking and problem solving via programming, designed for students interested in applying computation to the humanities and text analysis. It frames computation as a process of designing systematic solutions to problems; implementing, testing, and verifying those solutions; and making the solutions accessible to other scholars and investigators. Students learn fundamentals of computer programming using Python, including basic data structures, flow control structures, functions, recursion and elementary object-oriented programming, as well as discussion of higher-level concepts including abstraction, modularity, reuse, testing, and debugging. By the end of the semester, students develop an understanding of computational problem solving and gain experience implementing that problem solving in the context of text analysis. MOI: [QF]

DCS/MATH 203 Discrete Structures (revised from former catalog entry)
This course introduces students to discrete mathematical structures. In developing our toolkit for algorithmic solutions to systems problems, we will explore questions about the nature of events, change, uncertainty, and interconnectedness in natural, physical, and social systems. In and out of the classroom, we will engage actively with terminology, theoretical foundations, and strategies for developing and testing various mathematical and computational methods in various contexts. Students may complete assignments in a computing language of their choice and will be asked to compare computing languages. Mathematical topics include logic, proof, probability, and linear algebra, and will be studied through experiential, symbolic, numeric, visual, and verbal means against the backdrop of the complex, interconnected world we experience. MOI: [QF]

DCS 204 / AMST 205 Archives, Data, and Analysis
The computational humanities comprise a fast-growing and exciting field that is changing the way scholars work and think. This course provides an opportunity for students with some experience with programming to immerse themselves in semester-long projects in digital environments, moving from "analog" archives, through data structuring, and quantitative analysis, and culminating with a project that makes both the humanities and quantitative analyses legible for people from diverse backgrounds. MOI: [HS,SR]

DCS 206 The Past, Present, and Possible Dystopian Future of Computing
In this course students examine the history, present, and possible future of computing through film and literature, focusing on questions at the intersection of computing, digital studies, and communication: Who are the stakeholders and participants in this

intersectional area? What are the uses and abuses of data and computing in society? Who has the power of technology and who does not, and what are the consequences of that power? Recommended background: Prior critical-studies-oriented digital and computational studies course or similar course work in Africana, American studies, Latin American and Latinx studies and/or gender and sexuality studies. MOI: [CP]

DCS 211 Computing for Insight

Building on DCS 109/111, this course explores practical application of software composition as a bridge to other disciplines. Students continue to develop programming and problem-solving skills, with the clear purpose of providing insight to inquiry in other fields that is made possible by modern computing, software composition, and libraries. The course includes study of additional data structures and algorithms; data harvesting, analysis, and visualization; machine learning; modeling and simulation; and considerations of human- and machine-efficiency. As a final course project, students design, implement, and assess a computing project of their choosing. MOI: [QF]

DCS/HIST 212 Digital History Methods

Through a combination of analytical, experiential, and collaborative exercises, students merge traditional historical methods with digital tools to explore new useful methodologies for collecting, analyzing, and disseminating historical knowledge. They develop technical and theoretical proficiency within the broader field of digital humanities. They engage digital tools and resources to rethink old historical questions. They develop with new questions that can be investigated only through digital practice. They contemplate avenues for collaboration between historical research and public communities. Finally, they weigh the practical and theoretical implications of using digital history to create more inclusive scholarship. MOI: [HS]

DCS 229 Data Structures and Algorithms

This course provides an introduction to common data structures and selected algorithms for solving more complex problems. Topics covered include concrete data types (arrays and linked structures); abstract data types (including stacks, queues, trees, and maps); an introduction to fundamental algorithms including sorting, graph-search algorithms (breadth-first search, depth-first search), and greedy algorithms; and basic algorithm analysis (big-Oh). The course focuses on applying data structures and algorithms for problem solving, rather than on data-structure implementation details and formal analysis. MOI: [QF]

DCS/HIST 301C Public History in the Digital Age

Public history takes place beyond history classrooms and academic contexts. Traditionally, it has been found in museums, walking tours, and performances, and has told the stories of people with social and political privilege. Increasingly, however, public history has come to focus on a greater range of voices, and takes place in a wider range of forms: on websites, graphic novels, interactive sensory experiences, social media, and other digital spaces. In this community-engaged course, students learn to see public history "in the wild," engage with primary sources, and present those sources and historical interpretation to the public in digital form. Students with interests in history and public engagement are encouraged to enroll in this course. MOI: [HS]

DCS 307 Theory and Implementation of Computer Simulation Models
This course introduces topics in computer simulation, focusing on the underlying theory, implementation, and analysis of discrete-event simulation models. Topics include discrete-event simulation, Monte Carlo simulation, random number generation, discrete and continuous stochastic models, input modeling, statistics and visualization for output analysis, and point and interval parameter estimation in simulation contexts. The course focuses heavily on real-world systems that are appropriately modeled using queuing and agent-based simulation models. The course is simultaneously theoretical and computational. Students use mathematical and statistical derivations, as well as existing software libraries in R and Python, to understand and analyze simulation models. Software development is also a significant component of the course, as students work in teams to design, implement, and analyze the results of their own models. MOI: [QF]

DCS 317 Algorithms & Theory of Computation
This course presents an introduction to the design and analysis of efficient algorithms, and to the theory of computation. Students will learn to apply and analyze standard problem-solving techniques, and will also examine the spaces of problems that can be solved versus those that cannot be solved. Topics in algorithms will cover design strategies including divide & conquer and dynamic programming, as well as tools for analyzing and conveying algorithmic performance and correctness. The course will also include topics from the theory of computation, including P vs NP and reductions, deterministic finite automata, Turing machines, undecidability, and the halting problem. MOI: [QF]

DCS 320 Health Informatics and Digital Health: Data, Systems, and Innovations in Healthcare
Provides a comprehensive overview of health informatics, examining how digital health technologies and data-driven solutions are transforming healthcare. Includes study of design, implementation, and interoperability of health information systems, including electronic health records (EHRs) and clinical decision support systems (CDSS), and data standards. Students will learn to acquire, preprocess, and analyze health data using machine learning and predictive modeling techniques to enhance healthcare outcomes. Students will explore digital health innovations such as telemedicine, wearable devices, and mobile health applications. Students will critically analyze ethical, societal, and regulatory challenges, particularly in patient data privacy, security, and bias in artificial intelligence. Students will also develop and evaluate innovative digital health solutions to address current and future healthcare challenges through practical, project-based learning using real-world datasets. Recommended background: Basic knowledge of structured computer programming is necessary. Additional study in either computer science or in health sciences is recommended. MOI: [QF]

DCS 325 Introduction to Web Development
This course provides an introduction to full-stack web development, including user-facing website design and construction, back-end frameworks, and client communication. The course will cover fundamental core web technologies (HTML, CSS, and JavaScript), as well as select libraries and frameworks used in current practice (e.g., React, Tailwind CSS). The course will also cover various web data formats (e.g., JSON, XML), server-side / back-end frameworks (e.g., Firebase, Django, Flask), fundamental UI and UX concepts (e.g.,

prototyping, usability, accessibility), and website security. Students will work directly with clients, with a focus on planning and maintenance.

DCS 360 Independent Study
Students, in consultation with a faculty advisor, individually design and plan a course of study or research not offered in the curriculum. Course work includes a reflective component, evaluation, and completion of an agreed-upon product. Sponsorship by a faculty member in the program/department, a course prospectus, and permission of the chair are required. Students may register for no more than one independent study per semester.

DCS 375 Network Analysis
Networks are everywhere. They describe how people, organisms, and ideas connect and interact. Studying networks reveals patterns, systems, and frameworks that are, in many cases, otherwise invisible. This course introduces network analysis as a tool that offers insights into the construction of social, biological, and information systems. It scaffolds the terminology and theoretical underpinnings of network science. It also introduces the data wrangling, qualitative analysis, quantitative analysis, critical analysis, and data visualization tools that often accompany the studies of networks. Recommended background: Prior coursework in critical digital studies and R programming, data cleaning, and/or significant programming experience. MOI: [SR]

DCS S13 Computation & Mathematical Art
This course will explore art and how art can be used to visually demonstrate computational and mathematical concepts. Explore the cross-over between two very different disciplines and what you can do with this knowledge. This course will feature work by prominent mathematicians and computer scientists in the mathematical and computational art space. Additionally, students will leave the course with a completed art project of their devising that demonstrates a computational or mathematical concept and a beginner level knowledge of crochet. The course will also include a historical overview of how art, specifically weaving and knitting, were instrumental in creating examples of how to store information that would be used as computer science was developed.

DCS S33 Introduction to Web Development
This course provides an introduction to full-stack Web development, including user-facing website design and construction, back-end frameworks, and client communication. The course will cover technologies for client-side development (HTML, CSS, and JavaScript), various web data formats (e.g., JSON, XML), server- side web frameworks (e.g., Django, Flask), fundamental UI and UX concepts (e.g., prototyping, usability, accessibility), and website security. Students will work directly with clients, with a focus on planning and maintenance.

DCS S31 Human-Robot Interaction Design Workshop
How should robots co-exist with humans? In this course, you will learn about the field of human-robot interaction (HRI) with a focus on the theory and processes used to design robots that interact with people. You will work in teams to iteratively design, build, and evaluate a new human-robot interaction system. At the end of the term, each team will

share their findings and design process. No prior programming or robotics experience is required.

II. Descriptions of Courses Currently Cross-Listed with DCS by Other Units

(In alphabetical order by hosting unit, then by course number)

ECON/DCS 351 Computational Macroeconomics
This course is an introduction to dynamic general equilibrium models, which have become the workhorses of modern macroeconomics. These models involve intertemporal optimization by the different agents in the economy: households, firms, and the government. They are often used to analyze the modern theories of growth and aggregate fluctuations, and to study the role of monetary and fiscal policy. Most of these dynamic models, however, do not have analytical (closed form) solutions and one often has to rely on computational methods to analyze their behavior. The goal of this course is to provide an introduction to the computational tools that are necessary to solve dynamic economic models quantitatively. MOI: [QF, SR]

ECON/DCS 368 Data Science for Economists
Economics is at the forefront of developing statistical methods for analyzing data collected from uncontrolled sources. Because econometrics addresses challenges such as sample selection bias and treatment effects identification, the discipline is well-suited to analyze large or unstructured datasets. This course introduces practical tools and econometric techniques to conduct empirical analysis on topics like equality of opportunity, education, racial disparities, and more. These skills include data acquisition, project management, version control, data visualization, efficient programming, and tools for big data analysis. The course also explores how econometrics and statistical learning methods cross-fertilize and can be used to advance knowledge on topics where large volumes of data are rapidly accumulating. We will also cover the ethics of data collection and analysis. MOI: [QF]

GSS/DCS 319 The Future of Work at the Human-Technology Frontier
This seminar will explore privilege, power, place, and concepts of labor within digital economies of communication and information exchange. As digital technologies continue to blur the boundaries between leisure and work, surveillance and data collection become invisibilized and normalized processes. This class will combine methodologies from feminist research practices and critical digital studies while exploring the rapid coevolution of labor and technology. We will discuss place and transnational technological labor, unpack the black box of artificial intelligence and machine learning, and explore the digital spaces for activism towards an open and inclusive science. Students in this course will gain critical thinking and analytical skills in an interdisciplinary classroom setting that incorporates scholarship and methodologies from both humanities and STEM disciplines.

GSS/DCS S13 Infrastructures
Popular representations of digital technologies often present them as somehow independent of material constraints-as inherently clean, "green," and ethereal as a cloud. Those images belie the realities of the information economy's myriad environmental

impacts, from resource depletion to water pollution to massive energy consumption. This course, an introduction to the history and politics of infrastructure, directs attention to relationships between human and nonhuman nature, using everyday personal computing as a point of departure. Throughout, students engage with activists, regulators, and maintainers working toward justice and sustainability in the digital age.

MATH/DCS 117 Introduction to Data Science: Data Visualization
This course offers an introduction to data science through data visualization. Through hands-on assignments, students will develop their skills in data cleaning, analysis, and visualization using the R programming language and Github for version control. In the course students will learn to calculate and describe data using descriptive statistics and how to create a range of data visualizations to explore variation and covariation in data. Students will also learn to critique and reflect on data visualizations encountered in everyday life. No prior experience in data science or programming is necessary, making this course accessible to all students interested in exploring the dynamic field of data science. MOI: [QF]

MATH/DCS 316 PIC Math: Community Engaged Data Science
This PIC Math (Preparation for Industrial Careers in Mathematical Sciences) course is intended for students with a strong interest in industrial applications of mathematics and computation. Students work in teams on a research problem identified by a community partner from business, industry, or government. Students develop their mathematical and programming skills as well as skills and traits valued by employers of STEM professionals, such as teamwork, effective communication, independent thinking, problem solving, and final products.

MATH/DCS 331 Mathematics for Machine Learning
This course begins with linear regression models and introduces students to a variety of techniques for learning from data, as well as principled methods for assessing and comparing models. Topics include bias-variance trade-off, resampling and cross-validation, linear model selection and regularization, classification and regression trees, bagging, boosting, random forests, support vector machines, generalized additive models, principal component analysis, unsupervised learning and k-means clustering. Emphasis is placed on the mathematics behind these concepts and on the statistical computing in a high-level language (e.g. R or Python). MOI: [QF, SR]

MATH/DCS 355A Numerical Analysis
This course studies the best ways to perform calculations that have already been developed in other mathematics courses. For instance, if a computer is to be used to approximate the value of an integral, one must understand both how quickly an algorithm can produce a result and how trustworthy that result is. While students implement algorithms on computers, the focus of the course is the mathematics behind the algorithms. Topics may include interpolation techniques, approximation of functions, solving equations, differentiation and integration, solution of differential equations, iterative solutions of linear systems, and eigenvalues and eigenvectors. MOI: [QF]

MATH/DCS 355D Chaotic Dynamical Systems
The field of dynamical systems is best understood from both theoretical and computational viewpoints, as each informs the other. Students explore attracting and

repelling cycles and witness the complicated dynamics and chaos a simple function can exhibit. Topics include chaos in discrete versus continuous dynamical systems, bifurcations, and attractors, with applications to biology and physics. While there will be a significant computational component to the course, previous coding experience is not required. MOI: [QF]

MATH/DCS 355H Numerical Linear Algebra
This course studies the best ways to perform calculations that have been developed in Linear Algebra. Topics may include solving systems of equations, error and condition numbers, least squares, and eigenvalues and singular values.

MATH/DCS S45T Mathematical Image Processing
This course introduces mathematical methods in digital image processing, including basic image processing tools and techniques with an emphasis on their mathematical foundations. Students implement the theory using MATLAB. Topics may include image filtering, image enhancement, edge detection, and image segmentation.

MUS/DCS 219 Composing Sonic Systems
This course takes computational and communications systems concepts, such as randomness, probability, generativity, signal processing, feedback, control (and non-control), and listening as parameters for electronic sound composition. Using the free, user-friendly visual programming environment, Pure Data (Pd), students create unique software-based artworks and compositions. Creative projects are grounded in theoretical and historical readings as well as listening assignments that provide context for the application of computational concepts and communications systems thinking to sonic arts practice. The course culminates in a final showing of sound art installations and performances. Recommended background: experience in one or more of the following: music composition, music performance, experimental arts, digital media, computer programming, electronics, media studies. MOI: [CP]

NRSC/DCS 306 / PSYC 305 Animal Learning
The course examines historical and recent trends in animal learning. Topics include classical and operant conditioning, biological constraints on learning, and cognitive processes.

NRSC/DCS 357 Computational Neuroscience
The brain is a complex object, and studying it scientifically requires a facility with tools and concepts for analyzing high dimensional data. This course will provide a survey of such tools through representative case studies in perception (how many types of odors are there?), genomics (how do we classify cell types?), and neural coding and dynamics (how does brain activity encode attributes of the world?). Students will develop intuitions for framing fundamental neuroscience questions as data-driven problems, and will also develop skills for exploring, visualizing, modeling, and interpreting data. No prior experience with coding is assumed or expected, and the course will emphasize the use of high-level computational tools rather than implementation of algorithms from scratch. MOI: [SR]

PHIL 210 / DCS/NRSC 252 Philosophy of Cognitive Science

Cognitive science is the interdisciplinary study of the mind, including psychology, neuroscience, linguistics, computer science, and philosophy as its core. This course examines the conceptual foundations of cognitive science, and different approaches to integrating findings and perspectives from across disciplines into a coherent understanding of the mind. Students also consider issues in the philosophy of science, the nature of mind, self, agency, and implicit bias. MOI: [AC]

PHIL/DCS S29 Logic: Possibility, Proofs, and Paradox
Building on PHIL 195 (Introduction to Logic), students consider the relationship between logic and reasoning, learn about modal logic (the logic of possibility and necessity), Turing machines, and alternative logics, prove some surprising metalogical results, and puzzle through some logical paradoxes.

PHYS/DCS 216 Computational Physics
An introduction to computational methods for simulating physical systems, this course focuses on the numerical analysis and algorithmic implementation necessary for efficient solution of integrals, derivatives, linear systems, differential equations, and optimization. While the course presents a rigorous introduction to the numerical analysis underlying these techniques, the emphasis remains on practical solutions to important physical problems. Students solve problems across the wide range of applications of computational physics including astrophysics, biological population dynamics, gravitational wave detection, urban traffic flow, and materials science. No prior experience in programming is required, though students without a technical computing background are encouraged to take PHYS s10 before enrolling. MOI: [QF, SR]

PHYS/DCS S11 Modeling & Data Analysis in the Physical Sciences
How are models used in the physical sciences? How are models informed by data in different disciplines, and how do we analyze it? This course will provide students with an opportunity to explore the many ways that models are used to conduct scientific research, and gain insights into how models can support their academic and career goals. We will discuss a variety of models, techniques, and tools used throughout the physical sciences, and guide students to use them for their own research project. Students will learn data fundamentals as well as specific applications of these data-driven techniques to fields within the physical sciences. Lessons will be taught through a lecture and lab component, going over the concepts before letting students work through coding projects, enabling them to create and conduct their own data-driven research. This course has no prerequisites, and aims to bring together students from a wide variety of academic disciplines and any level of coding experience.

THEA/DCS 170 Introduction to Digital Media
This introductory course explores the ever-evolving world of digital media in the performing arts, where technology, creativity, and communities converge. Students will be trained on the holistic and collaborative process from storyboarding to technical execution, specifically as it relates to live entertainment. We'll examine the history, current landscape, and emerging technologies in projection and video design. MOI: [CP]

III. Descriptions of Currently Planned Courses in DCS

DCS 2xx Human-Computer Interaction Design
We are constantly interacting with and through computers. This course is an introduction to the field of human-computer interaction, with a focus on human-centered design. In this course students will learn theory behind the design and evaluation of interactive computer systems, and put what they learn into practice. Students will work in groups to analyze, prototype, and evaluate interactive system designs. This course is open to students who have completed one DCS course.

DCS 2xx Humanities Data Lab
Humanists often turn to data as part of a toolkit that allows us to tell stories that are obscured or absent from traditional archives. In this class, students learn to critically read and interpret humanities datasets, and how to bring digital humanities lenses to bear on those datasets, in concert with theories and methodologies across the humanities. This course is open to students who have completed any of the DCS 109 set.

DCS 3xx Computers Reading Humans; Humans Reading Computers
This course introduces students to the principles and practices of computational text analysis. In the process, it asks students to consider what insights we might derive from reading texts at scale, and what we might loose from those same methodologies. This course is open to students who have completed DCS 204, DCS 211 or DCS 229.

DCS 3xx Introduction to Robotics
What are robots? How are they controlled? Are they going to take over the world? In this course, students gain hands-on programming, problem-solving, and design experience as they learn about robots. This course focuses on the computational side of robotics — including robot control algorithms, systems programming, and computational feasibility — and how these computer systems interface with society. This course is open to students who have completed DCS 211 or DCS 229.

DCS 3xx The Hierarchy of Computer Systems
This course provides an overview of the hierarchy of abstractions that comprise modern computer systems, from the machine level to program abstractions to networked and internet communication. At the machine level, the course covers important aspects of data representation, basic computer architecture, file systems, and operating systems. The course also introduces fundamental elements of running a computer program, including program compilation, translation, and execution. Finally, the course explores fundamental notions of computer networking, including layer architectures, communication protocols (TCP/IP), internet communication, and network security.

DCS 4xx Design Capstone
In this course, students work in teams to design, build, analyze, and critique a software/digital solution to a research problem identified by a community partner from academia, non-profit, industry, or government sectors. Throughout this culminating experience, students will leverage their digital and computational skills developed across the major, including those skills from computer science, data science, critical digital

studies, human-centered design, and community-engaged learning. Students will continue to build skills in teamwork, effective communication, independent and critical thinking, problem solving, and project management. Students will use multiple forms of communication consistent with the discipline, including storyboard design plans, oral presentations, software/digital artifacts, and supporting documentation, requiring regular revision throughout as the community partners provide feedback. Students will also be expected to continue to develop their own individual portfolios as a way to communicate their skills and experiences. Partners will vary by offering. (This course will be tagged as Community Engaged Learning and as a Purposeful Work Infusion Course.)