“Digital Intelligence”: A New Undergraduate Certificate Track

Introduction:

The Digital Intelligence certificate is a new track within the Science & Society undergraduate certificate program. This new track focuses on providing undergraduate students with foundational understanding of contemporary and emerging computational thinking, such as artificial intelligence, cybersecurity, web/mobile technologies, databases and software engineering. This training in technology will be coupled with requisite ethical, legal, social, and policy frameworks needed to understand the complexities of technology’s impact on our world. The Digital Intelligence track provides students with a holistic understanding of revolutionary technologies and the responsibility researchers, designers, and regulators bear in ensuring they are built and implemented responsibly.

This certificate track builds on wide-ranging partnerships with other units across Duke. The curricular contents draw from Computer Science, Electrical and Computer Engineering, as well as other units spanning Arts & Sciences, Engineering, Public Policy, and Law. The co-curricular contents draw from the Innovation Co-Lab and the + Data Science program. The Capstone experience draws from the Data+ program pioneered by the Rhodes Information Initiative at Duke (iiD) as well as partner programs like Code+, C5+, and Bass Connections. Science & Society and its partners above are key players in Duke’s overall framework for training in computation and its societal implications, with that overarching framework led by the Duke Center for Computational Thinking.

Program Details:

The core and four elective areas of the Digital Intelligence certificate track provide a broad examination of computational technology as well as ethics, policy, and communication issues. The capstone provides an opportunity for experiential learning and potential real-world impact. The certificate is targeted to all Duke students.

To complete this certificate track, students will complete the core and capstone requirements (two courses), plus 4 courses drawn from the following elective areas, with at least one from each of the first three areas. Up to one elective course can be fulfilled instead with 50 hours of co-curricular learning activities (more detail on this below). Below we summarize the program, and in the Appendix more details are provided on the classes.

Core Class:

- **Computation, Ethics, and Policy (SCISOC 256-02):** This course offers students an introduction to computational artificial intelligence theory and techniques and the corresponding ethical considerations for developing and using these tools.
Elective Area 1: Computational Thinking
Elective Area 2: Ethics and Policy
Elective Area 3: Communication
Elective Area 4: Computation in Practice

Co-Curricular Option: Elective Area 4 can be fulfilled by 50 hours of in-person and/or virtual learning opportunities across 4 years to dive deeper into relevant areas. Innovation Co-Lab Roots (https://colab.duke.edu/roots) and + Data Science (https://plus.datascience.duke.edu/) sessions are a primary means of fulfilling these co-curricular learning requirements, but there are many others as well. The hours will be logged to ensure participation.

Capstone:
- **Computing, Data, and Ethics:** The capstone is a seminal course that will be offered to rising and existing fourth-year students upon completing the Core and Elective courses. The course will be designed as an experiential project-based class, whereby student groups engage in either summer-long or semester-long partnerships with real-world clients to examine ethical technology issues with computational thinking. Participation in programs such as Data+, Code+, CS+, and approved Bass Connections projects will satisfy the Capstone course requirement, with a culminating experience of a reflection paper, poster, or presentation that reflects on those experiential programs in the senior year.

Summary

Unprecedented access to data, computing, and technology is transforming our world and our concept of a liberal arts education. Regardless of undergraduate discipline, all Duke University students’ futures will be influenced by the interrelationship between computation, data, technology, industry, governments, and society. The proposed Digital Intelligence track within the Science & Society certificate represents Duke’s commitment to interdisciplinary pedagogical frameworks that prepare students for a complex future. Students will access robust, thoughtful, and critical considerations of technology and its implications for individuals and society.

Appendix: Detailed Course Planning

The following lists the recommended courses for each elective area. These lists will be updated regularly, and other courses may satisfy the requirement with permission of the Certificate Director.

Elective Area 1: Computational Thinking

*Note: For students in traditional computing majors (ECE/CS), this elective can be satisfied by any course with a significant programming component that is used to satisfy their major requirement. The list of courses below are examples of what are most relevant to this track.*
For students starting with little or no prior programming experience or computational background:

- **AI for Everyone**: Building upon the core class, this course will offer students a robust overview of core (applied) artificial intelligence development techniques.

- **Interdisciplinary Computing (CompSci 102L/Neuro 104L)**: Computer science connects to multiple other disciplines, from engineering to the natural sciences, social sciences, and humanities. This course provides an introduction to the practices and principles of computer science and programming and their impact on and potential to change the world. Assignments will be motivated by problems drawn from a) the engineering of affective computing; b) the natural science of neuroscience, and c) certain topics in the humanities. Programming will involve Python as well as appropriate libraries, and APIs to process, analyze, and visualize data. Students will master design, implementation, and analysis emphasizing abstraction, encapsulation, and problem decomposition. No previous programming experience required. This course satisfies the prerequisite for CompSci 201.

- **Foundations of Data Science (CompSci 116)**: Given data arising from some real-world phenomenon, how does one turn that data into knowledge and that knowledge into action? Students will learn critical concepts and skills in computer programming and statistical inference in the process of conducting analysis of real-world datasets. Students will write computer programs for projects using the Python programming language. In considering applications, we will discuss how data can be used responsibly to benefit society. No previous programming experience required. This course satisfies the prerequisite for CompSci 201.

- **Computational Methods in Engineering (EGR 103L)**: Introduction to computer methods and algorithms for analysis and solution of engineering problems using numerical methods in a workstation environment. Topics include; numerical integration, roots of equations, simultaneous equation solving, finite difference methods, matrix analysis, linear programming, dynamic programming, and heuristic solutions used in engineering practice. This course does not require any prior knowledge of computer programming.

- **Intro to Global Health Data Science (STA 198)** - Name changing from Intro to Health Data Science as of fall

  Rigorous introduction to health data science using current applications in biomedical research, epidemiology, and health policy. Use modern statistical software to conduct reproducible data exploration, visualization, and analysis. Interpret and translate results for interdisciplinary researchers. Critically evaluate data-based claims, decisions, and policies. Includes exploratory data analysis, visualization, basics of probability and inference, predictive modeling, and classification. This course focuses on the R computing language. No statistical or computing background is necessary. Not open to students who have taken a 100-level Statistical Science course, Statistical Science 210, or a Statistical Science course numbered 300 or above.

- **Introduction to Data Science and Statistical Thinking (STA 199)**

  Short name: Intro to Data Science

  Learn to explore, visualize, and analyze data to understand natural phenomena, investigate patterns, model outcomes, and make predictions, and do so in a reproducible and shareable manner. Gain experience in data wrangling and munging, exploratory data analysis, predictive modeling, and data visualization, and effective communication of results. Work on problems and case studies inspired by and based on real-world questions and data. The course will focus on the R statistical computing language. No statistical or computing background is necessary. Not open to students who have taken a 100-level Statistical Science course, Statistical Science 210, or a Statistical Science course numbered 300 or above.
from the courses above or through co-curricular offerings):

- **Data Structures and Algorithms (CompSci 201):** Analysis, use, and design of data structures and algorithms using an object-oriented language like Java to solve computational problems. Emphasis on abstraction including interfaces and abstract data types for lists, trees, sets, tables/maps, and graphs. Implementation and evaluation of programming techniques including recursion. Intuitive and rigorous analysis of algorithms. Prerequisite: Computer Science 101 or Engineering 103L, or equivalent.

- **Everything Data (CompSci 216):** Data is the new currency. In every walk of life, people leave digital traces, which are stored and analyzed at both individual and population levels, by businesses for improving products and services, by governments for policy-making and national security, and by scientists for advancing the frontiers of human knowledge. This course serves as an introduction to various aspects of working with data—acquisition, integration, querying, analysis, and visualization—and data of different types—from unstructured text to structured databases. Through lectures and hands-on labs, the course covers both fundamental concepts and computational tools for working with data and applies them to real datasets in a capstone team project. Recommended prerequisite: programming in Python.
• **Fundamentals of Data Analysis and Decision Science (EGR/Math 238L):** This course provides a mathematically rigorous and broad foundation for key concepts in probability and statistics, as well as the application of probability and statistics to the mathematical modeling of non-deterministic systems. The main motivation of the course is to show how these concepts are fundamental to a variety of current data analysis techniques, and to demonstrate applications of these techniques in situations relevant to all engineering majors. Prerequisite: (Mathematics 216, 218, or 221) and (Engineering 103L or Computer Science 101L).

• **Programming, Data Structures, and Algorithms in C++ (ECE 551D):** Students learn to program in C and C++ with coverage of data structures (linked lists, binary trees, hash tables, graphs), Abstract Data Types (Stacks, Queues, Maps, Sets), and algorithms (sorting, graph search, minimal spanning tree). Efficiency of these structures and algorithms is compared via Big-O analysis. Brief coverage of concurrent (multi-threaded) programming. Emphasis is placed on defensive coding, and use of standard UNIX development tools in preparation for students' entry into real-world software development jobs. Not open to undergraduates. Instructor: Hilton. 3 units.

For students with more advanced computational backgrounds:

• **Computer Architecture (CompSci/ECE 250D):** Computer structure, assembly language, instruction execution, addressing techniques, and digital representation of data. Computer system organization, logic design, microprogramming, cache and memory systems, and input/output interfaces. Prerequisite: Computer Science 201.

• **Introduction to Mobile Programming (CompSci 290)**
• **Web Application Development (CompSci 290)**
• **Algorithms in the Real World (CompSci 290)**
• **Introduction to Computational Imaging (CompSci 290)**
• **Reinforcement Learning (CompSci 290)**
• **Software Design and Implementation (CompSci 307/308)**
• **Introduction to Operating Systems (CompSci 310)**
• **Introduction to Database Systems (CompSci 316)**
• **Introduction to the Design and Analysis of Algorithms (CompSci 330)**
• **Computer Security (CompSci 351)**
• **Computer Network Architecture (CompSci 356)**
• **Introduction to Artificial Intelligence (CompSci 370)**
• **Elements of Machine Learning (CompSci 371)**
• **Delivering Software: From Concept to Client (CompSci 408)**
• **Data Science Competition (CompSci 474)**
• **Computer Vision (CompSci 527)**
• **Algorithmic Aspects of Machine Learning (CompSci 590)**
• **Machine Learning (CompSci 671)**
• **Machine Learning and Data Mining (STA 325)**
• **Bayesian Inference (STA 360)**
• **Sta. Learning and Inference (STA 432/MATH 343)**
• **Topological Data Analysis (MATH 412)**
• **Introduction to High Dimensional Data Analysis (MATH 465/COMPSCI 445)**
• **Math of Machine Learning (MATH 466)**
- **Introduction to Robotics and Automation (ECE 383):** Fundamental notions in robotics, basic configurations of manipulator arm design, coordinate transformations, control functions, and robot programming. Applications of artificial intelligence, machine vision, force/torque, touch and other sensory subsystems. Design for automatic assembly concepts, tools, and techniques. Application of automated and robotic assembly costs, benefits, and economic justification. Selected laboratory and programming assignments. Prerequisites: ECE 280L or EGR 224L.

- **Engineering Software for Maintainability (ECE 458):** Students will refine their ability to engineer software, with a focus on maintainability. Teams complete a large programming project whose requirements undergo evolutions during the semester. Prior to each evolution, teams submit not only their code for the current requirements, but also a written document analyzing their current design and reflecting on how past design choices impacted their most recent work. Teams must consider appropriate engineering standards and realistic constraints. Prerequisite: Computer Science 307D or 308; ECE 230L, ECE 250D, ECE 270DL, ECE 280L; Mathematics 353; Physics 152L; Chemistry 101DL; and (STA 130 or MATH 230 or ECE 555 or ECE 380 or STA 240L or EGR 238L or MATH 340). Instructor: Bletsch or Hilton.

- **Applied Probability for Statistical Learning (ECE 480):** This course discusses topics in Bayesian probability and its application to foundations of statistical learning. The primary objectives of the course are to provide a mathematically rigorous foundation in Bayesian probability and inference, develop strong intuition for Bayesian constructs, provide a foundation in statistical learning, and to show how Bayesian methods are fundamental to a variety of modern statistical learning techniques. Topics include probabilistic reasoning, Bayesian inference, linear models, mixture models, and model selection. Pre-reqs: (MATH 216 or MATH 218 or MATH 221) & (EGR 238L or MATH 230 or ECE 380 or ECE 555 or STA 240L or Math 340) & (EGR 103L or COMPSCI 101L).

- **Fundamentals of Computer Systems and Engineering (ECE 550D):** Fundamentals of computer systems and engineering for Master's students whose undergraduate background did not cover this material. Topics covered include: Digital logic, assembly programming, computer architecture, memory hierarchies and technologies, IO, hardware implementation in VHDL, operating systems, and networking. Undergraduates may not take this course, and should take ECE 250D, ECE 353, and/or ECE 356 instead. Instructor: Hilton. 3 units. Co-requisite: ECE 551D.

- **Mobile Application Development (ECE 564):** Explores the world of mobile application development with focus on needs of engineers. Centered on Apple environment, with the development environment being on OS X and the target environment being an iOS device – iPad, iPhone, iPod Touch or Apple Watch. Real world context focused on the common programming “patterns” for engineers in academia or business – standalone apps, apps connected to other systems, apps connected to the cloud. Covers fundamentals essential to understanding all aspects of app development. Taught in a team environment. Students required to present their project proposals and deliver an app as a final project. Prerequisites: CS 308 or ECE 551D. Instructor: Staff.

- **Introduction to Machine Learning (ECE 580):** Introduction to core concepts in machine learning and statistical pattern recognition, with a focus on discriminative and generative classifiers (nearest-neighbors, Bayes, logistic regression, linear discriminant, support vector machine, and relevance vector machine). Dimensionality reduction and feature selection. Classifier performance evaluation, bias-variance tradeoff, and cross-validation. Prerequisite: (Mathematics 216, Mathematics 218, Mathematics 221, or ECE 586) and (Computer Science 201 or ECE 551D) and (ECE 480 or ECE 581). Not open to students who have taken Computer Science 671D
• **Comp Eng ML & Deep Neural Nets (ECE 590):** This course examines various computer engineering methods commonly performed in developing machine learning and deep neural network models. The focus of the course is on how to improve the training and inference performance in terms of model accuracy, size, runtime, etc. Techniques that are widely investigated and adopted in industrial companies and academic communities will be discussed and practiced. Programming practices on these techniques are designed with heavy utilization of the PyTorch package. Classes will be running by combining lecture sessions and discussion sessions for the programming practices. Prerequisite: Computer Science 201.

• **Natural Language Processing (ECE 684):** Introduction to algorithmic and analytic methods specific to textual data. Subject areas covered are speech recognition, optical character recognition, text parsing, and document analysis. Analysis tools taught include sentiment analysis/topic models, auto-correct, auto-complete, and translation systems. Applications to brain-computer interface communication systems, intelligent personal assistants, and plagiarism detection systems. Prerequisite: STA 130L, STA 240L, Mathematics 230, Mathematics 340, ECE 380, ECE 555, ECE 580, ECE 581, ECE 682D, EGR 238L, or ECE 551D.

• **Deep Learning (ECE 685D):** Provides an introduction to the machine learning technique called deep learning or deep neural networks. A focus will be the mathematical formulations of deep networks and an explanation of how these networks can be structured and “learned” from big data. Discussion section covers practical applications, programming, and modern implementation practices. Example code and assignments will be given in Python with heavy utilization of PyTorch (or Tensorflow) package. The course and a project will cover various applications including image classification, text analysis, object detection, etc. Prerequisite: ECE 580, ECE 681, ECE 682D, Statistical Science 561D, or Computer Science 571D.

**Elective Area 2: Ethics and Policy**

• **Digital Intelligence (offered Spring/Fall) (SCISOC 585-01):** The Digital Intelligence course helps students navigate and understand and analyze the ethical and social impact of emerging technologies through an applied ethical lens. In a flipped-classroom format, students will watch asynchronous videos on a weekly basis featuring leading technology, ethics, and policy experts as they discuss relevant and timely topics such as algorithmic bias, the impact of social media on democracy, and privacy in the digital age. Students will meet weekly in small discussion groups to work through case studies and to critically engage with a practical ethics approach to the topics presented in the video and additional assigned material.

• **Intro. Cyber Policy (PubPol 290):** Cybersecurity is a global challenge and involves every aspect of our lives, our economy, and our national security. Although cybersecurity risks have been present for more than 30 years, over the last five years the risks have evolved along with the impact to the confidentiality, integrity and availability of data. Today there are substantial and growing concerns about individual safety from cybersecurity risks. To be successful, experts in technology must understand public policy, and experts in public policy must have foundational knowledge of technology. Collaboration among technologists, lawyers, public policy experts, economists, business leaders, and risk management professionals is critical to allow society to take advantage of the opportunities that internet connectivity provides, while mitigating the
risks created for organizations and individuals. This course introduces students to the fundamentals of organizational responsibilities, and political implications of “cyber policy” with a primary focus on cybersecurity and privacy. The class provides an overview of technologies, threats, public policy organization roles and responsibilities, existing legal frameworks, and opportunities to influence public policy to achieve better results for society.

- **Race, Gender, Class, & Computing (CompSci 190):** This course provides an introduction to social identity groups such as race, ethnicity, gender, class, sexuality, and ability. Diversity, equity, and inclusion (DEI) challenges impacting them (such as intersectionality, racism, bias, and discrimination) are discussed, including various social justice movements. Identity is then explored in the context of DEI challenges in computing/tech environments, technologies developed, and their impact on people from various identities (including policies and organizations dedicated to addressing them). No previous experience required.

- **Technical and Social Analysis of Information and the Internet (CompSci 342):** The development of technical and social standards governing the Internet and information technology in general. The role of software as it relates to law, patents, intellectual property, and IETF (Internet Engineering Task Force) standards. Written analysis of issues from a technical perspective with an emphasis on the role of software and on how standards relate to social and ethical issues. Current events as a driver for writing in traditional and online formats related to technology and policy. Open only to students with declared Computer Science major. Prerequisite: Computer Science 201.

- **Information, Technology, Ethics and Policy (PUBPOL 372, cross-listed in PJMS and POLISCI):** This is a course about the politics and policy surrounding information and technology. It will focus on how politicians, policymakers, economies, citizens and society watchers talk about, worry about and understand the internet and other technologies, such as phones, networks, etc., as well as uses, artificial intelligence, social media and more. The course will address information privacy and security, global information flows, technology in campaigns and elections, how technology has changed journalism and news, and how our identities are changed by our media use. Students will research, analyze and write about technology policy issues.

**Elective Area 3: Communication**

- **Science Communication and Data Visualization:** This course will offer students the ability to engage with science communication theory and practice while learning data-driven visualization with the goal of broader public engagement and communication on topics of science, technology, and data.

- **News as a Moral Battleground (PUBPOL 371):** Ethical inquiry into journalism and its effect on public discourse. Issues include accuracy, transparency, conflicts of interest and fairness. Topics include coverage of national security, government secrecy, plagiarism/fabrication, and trade-offs of anonymous sourcing.

- **Communication for Scientists (BIOETHIC 702-01):** This course provides students in the science with practical training in the communication of scientific research to non-scientists and helps them develop skills essential to doing meaningful outreach. Topics covered include the empirical benefits of communicating science and technology, speaking, writing, and storytelling practices for diverse audiences; answering difficult, controversial, and critical questions from the media;
and tweeting, blogging, and presenting research to engage non-scientists (including the lay public and policy-makers).

- **Communicating Science & Bioethics (BIOETHIC 502S-01):** Examination of the challenges and best practices for communicating scientific and bioethical issues to the public, journalists, and policymakers. Explores historical and cultural factors that influence public understanding of and attitudes toward scientific and bioethical issues. Students will draw on communication case studies from a variety of disciplines (genetics, neuroscience, law, bioethics) and their own academic interests as a context for developing writing and speaking skills essential for clear communication of complex topics to non-specialists.

- **Science & the Media (BIOETHIC 510S-01):** Those who write about science, technology, health and related policy must make complex, nuanced ideas understandable to the nonscientist in ways that are engaging and entertaining, even if the topic is far outside the reader’s frame of reference. Course examines different modes of science writing, the demands of each and considers different outlets for publication and their editorial parameters. Students interview practitioners of the craft. Written assignments include annotations of readings and original narratives about science and scientists. Course considers ways in which narrative writing can inform and affect policy. Prerequisites: a 200-level science course and/or permission of the instructor.

**Elective Area 4: Computation in Practice**

- **Introduction to Computational Genomics (CompSci 260):** A computational perspective on the exploration and analysis of genomic and genome-scale information. Provides an integrated introduction to genome biology, algorithm design and analysis, and probabilistic and statistical modeling. Topics include genome sequencing, genome sequence assembly, local and global sequence alignment, sequence database search, gene and motif finding, phylogenetic tree building, and basic gene expression analysis. Methods include dynamic programming, indexing, hidden Markov models, and elementary supervised and unsupervised machine learning. Development of practical experience with handling, analyzing, and visualizing genomic data using the computer language Python. The course will require students to program often in Python. Students coming in to the course must already know how to program in some computer language, but it need not be Python.

- **Computational Microeconomics (CompSci 323):** Use of computational techniques to operationalize basic concepts from economics. Expressive marketplaces: combinatorial auctions and exchanges, winner determination problem. Game theory: normal and extensive-form games, equilibrium notions, computing equilibria. Mechanism design: auction theory, automated mechanism design. Prerequisite: at least one of the following: Computer Science 230, 200-level Mathematics course, or 200-level Statistical Science course.

- **Computational Sequence Biology (CompSci 561):** Algorithmic and computational issues in analysis of biological sequences: DNA, RNA, and protein. Emphasizes probabilistic approaches and machine learning methods, e.g. Hidden Markov models. Explores applications in analysis of
high-throughput sequencing data, protein and DNA homology detection, gene finding, motif discovery, comparative genomics and phylogenetics, genome segmentation, DNA/RNA/protein structure prediction, with a strong focus on algorithmic aspects. Prerequisites: basic knowledge of algorithmic design (COMPSCI 330 or equivalent), probability and statistics (STA 611 or equivalent), molecular biology (BIO 201L or equivalent), basic computer programming skills (preferred programming languages: Python, Java, C/C++, Perl, R, or Matlab).

- **Seminar on Computational Biology (CompSci 590):** This seminar course focuses on topics in computational biology. We will emphasize themes that unite algorithms, modelling, and experimental results. Topics will include algorithms, modeling, and experimental validation for several areas, including protein design, protein: protein interactions, structural biology, structural immunology, and structure-based drug design.