School of Engineering

Department of Computer Science

Computer Science Courses for Non-Majors

The Department of Computer Science offers a wide array of courses for majors. But computer science systems affect all of us every day. Therefore, the CS department also offers a number of courses meant for non-majors, ranging from “Cyber for Future Policymakers,” an exploration of cyber technologies that policymakers need to understand, to “Computing for Developing Regions,” a study of technologies that can be used in the context of marginalized communities, to “Privacy in the Digital Age,” a course examining privacy in our brave new digitized world. Take a look! These courses may be of interest regardless of your major.

EN 0001-01: Introduction to Computational Design

With the availability of increased computing power, many engineering disciplines now rely on utilizing computation to explore different design options. We will learn how that is done: how to model a problem with math and how to use computers to optimize a product using that model. We’ll optimize all sorts of products from different disciplines. We will learn to program in MATLAB, and then use it to learn fundamental concepts such as a solution space, design-decision variables, constraints, optimal points within the design space and searching a design space using efficient algorithms.

EN 0001-08: Simple Robotics

This introduction to an engineering course applies programming (in Python) to the field of robotics, exploring engineering design and using code to read sensors, make decisions, and control robots. Students will learn to design and build their own mechanical solutions (using the LEGO Education Robotics platform, so accessible to anyone), and then develop their own algorithms and control solutions for using simple code to solve open-ended challenges. From simple drawing robots to more interactive exhibits, this course introduces the field of CS and the role of programming in automation and control. Acceptable for students with no prior coding experience, but still fun even to those who've

programmed before; come learn how coding can be used to manipulate the world around you!

EN 0001-18: Exploring Computer Science

Basic principles of computer science for students with minimal or no prior programming background. Fundamentals of design, coding, and testing computer programs. Fundamental algorithms for sorting and searching. Programming projects employ and demonstrate common algorithms. Projects analyze and visualize data from real applications. This sampling of various topics will give the student a taste of not only what constitutes computer science, but also a deeper understanding of mankind's most powerful tool. The course will prepare the student to take CS 11 in a subsequent semester, if so desired.

*Note: Section 18 should **NOT** be taken in the same semester as CS 11. (Diane Souvaine, Computer Science)*

ES 0002: Introduction to Computing for Engineering

All fields of engineering are becoming more reliant on computation, as disciplines develop the ability to quickly collect lots of real-world data and rely on accurate and efficient processing to make engineering design decisions. This introductory course starts with the basics of programming, learning the basic coding concepts, and then builds to have students plan and implement both small and medium-scale projects in various fields of engineering. With a focus on data science (the acquisition, processing, and visualization of data), students develop confidence and skills for modeling physical systems in code and applying computation to engineering design.

DS 153: Artificial Intelligence: Algorithms, Ethics, and Policy

Artificial intelligence algorithms are designed to assist human decision-making processes ranging from driving, medical diagnoses, and language translation to criminal sentencing, drone targeting, and fraud detection. Each application area where machine learning is applied raises complicated ethical and legal issues of bias, oversight, privacy, accountability, and liability for these algorithms and the resulting automated decision-making they implement. This course aims to introduce students to the technical underpinnings of artificial intelligence so that they can better understand the range of policy options for addressing these issues. Many policy documents and approaches for AI center on the development of vague, high-level principles for the technology (e.g., AI should be accountable, ethical, equitable, explainable, etc.) but offer very little by way of concrete guidelines or specific constraints for how automated decision-making systems should be designed or deployed. This course is predicated on the idea that the only way to operationalize these principles in any meaningful or clear way in a policy context is to actually understand some of the technology underlying machine learning systems.

CS 10: Computer Science for All

Computers are indispensable tools for research. This holds not only for more technical fields such as physics or chemistry but also for humanities and the social sciences. While most students are competent users of standard software such as word processing or spreadsheets, the real power of the computer is unleashed when we are able to program it ourselves and make it do exactly what we want it to do.

This course is aimed at people who want to learn how to use computer science to solve basic information processing problems, such as analyzing text data and performing elementary statistics on them. It will cover elementary principles of computer science and will teach the student to independently write their own programs in the computer language Python.

CS 11: Introduction to Computer Science

The study of computer science centers on two complementary aspects of the discipline. First, computer science is fundamentally concerned with the problem-solving methodologies it derives from its foundational fields: the design principles of engineering, mathematical theory, and scientific empirical study. Second, these methodologies are applied in the complex context of a modern day computing system. In this course we will address both of these important aspects. As a means for developing your design skills, we will discuss the fundamental features of a high level, general purpose programming language -- namely C++ -- and learn how to use it as a tool for problem solving. We will also consider the performance of solutions, and how to apply both analytical and empirical assessment techniques. Finally, we will explore the Unix operating system as a context for problem solving.

CS 14: Emerging Scholars Program (ESP)

ESP is a semester-long program with weekly hour-long workshops. The goal of the program is to encourage more students to pursue Computer Science beyond the introductory level and into the major by creating a program that encourages active participation and discussion of Computer Science related topics in a positive, relaxed, and open environment. In each workshop, a peer leader presents a set of problems from a specific Computer Science field, and the students work as a group to come up with algorithmic solutions and/or analyses of the problems. There is no programming and there are no homework assignments, just the self-contained workshops that demonstrate that Computer Science is a collaborative activity that is focused more on problem solving than on programming. Since there is no work outside of class, attendance at all sessions is mandatory and participation is required from every student. Students should be enrolled concurrently in either COMP 10 or COMP 11. Please see: https://esp.cs.tufts.edu/ for specific details.

CS 12: Cyber for Future Policymakers

Policymakers face decisions involving new technologies all the time—but they often have no idea what these technologies do and how they work. This

course covers those technologies that policymakers use every day—and the ones on the near horizon—including: internet architecture and basic networking, the Web, cloud architectures, cryptography, security and privacy, AI and machine learning, and open-source systems. Developing technologies, including quantum computing and post-quantum cryptography. Learn how they work and learn how to explain them in simple terms. The course includes a focus on communicating with policymakers (e.g., writing policy briefs). All that's needed is a programming course equivalent to COMP 10 or COMP 11.

CS 13: How Systems Work

How modern computing systems work, their power and their limits: How do apps represent information? What can a physical computer actually do (and not do), and how can a single device show web pages, play games, edit documents, track your calendar, etc., etc.? How do computer networks allow us to transfer information from place to place? Learn how systems work and be able to amaze your friends, your boss, and even a Congressional staffer as you explain how networks and IP communications actually happen. All that's needed is a programming course equivalent to COMP 10 or COMP 11.

CS 27: How Systems Fail

Most of our courses in Computer Science are about making things work: writing good code, choosing efficient algorithms, and generally understanding how systems work. Not this course! This course is about breaking things. Why is that interesting? Aside from being fun (who doesn't like to break things?), looking at how systems fail tells us a lot about how they work. We will look at the way in which failures manifest themselves and the resulting consequences, in particular with respect to cybersecurity. Failures also tell us a lot about the human processes behind systems design and implementation, and how those processes can fail. The course is designed for students with a basic technical background; no significant programming experience is necessary.

CS 28: Cybersecurity and Cyberwarfare

This course is fundamentally about the intersection of technology and politics. It is designed for both technical and non-technical students. Cybersecurity concerns the technical constraints of computer networks and software that comprise cyberspace. Cyberwarfare refers to the use of cyberspace by a range of actors—dissidents, terrorist groups, small states, and great powers—to gather intelligence, assert influence, project power, and conduct covert operations in the increasingly ambiguous area between peace and war. For non-technical students, the course provides an overview of how cyberspace works, with a focus on exploit methods including packet analysis, password cracking, social engineering, reconnaissance, and malware analysis. For technical students, the course provides an overview of the politics of cyberspace, with a focus on espionage, counterintelligence, offensive cyber operations, Internet governance, and the difficulties of establishing international norms.

CS 116: Introduction to Security

In this infamous and venerable course, students will receive a broad and holistic perspective on cyber security. While this may be looked at as a weakness as students don't master anything, it gives students a real and professional view of how interdisciplinary Cyber Security as a field is. Cyber Security is now an international crisis. Understanding the basics of Cyber Security is necessary to students of international security and political science. Yet, it is impossible to have a meaningful conversation on Cyber Security without knowing the fundamental technicals of the topic. Students will learn by doing. This is how most successful Security practitioners learned their craft. Cyber Security is a very hands-on field. One can talk all about Cyber Security, but the person will not be treated seriously without hands-on technical knowledge. The labs include analyzing network traffic for credentials sent in plaintext, performing reconnaissance on a target remote computer, finding vulnerabilities on a remote server, cracking passwords, breaking into a website and stealing data, playing Capture the Flags, and analyzing a live piece of malware. This course shifts away from the typical mindset of programmers/developers/software engineers. Software developers and

engineers are trained to write software based on customer requirements, functionality, and often never thinking about security. Students taking this course think about what could possibly go wrong, breaking things, abuse cases, violating invariants. Students who want to take this course need to have some computer programming experience, ideally have taken both the Introduction to Computer Science and Data Structures courses.

CS 150: Ethical Issues in Computer Science

Most of the topics covered in the course are things of relevance to any user of technology, or anyone living in a society in which computing plays a key part—things like the use of algorithms to make hiring or criminal decisions, facial recognition systems and policing, and data-driven medical treatments. Most of the salient features of these and other topics can be understood without deep technical expertise in how the systems are built, since many of the most pressing questions revolve around how they are put to use. Non-majors are welcome, as anyone willing to discuss and write about ethical and social issues will be a valuable contributor.

CS 151: Computing for Developing Regions

This course is for anyone who has interest in seeing an example of CS + X -- specifically, how technology can be used in the context of developing regions. This is an interdisciplinary, applied course, with an emphasis on understanding and solving problems that are specific to developing regions. A wide range of case studies will expose students to a range of problems -- from healthcare to gender issues -- and how different tools from computer science can be applied to potentially solve them.

CS 182: Cyber in the Civilian Sector

This class is a survey of non-military cyber policy issues including Internet shut-downs, online censorship, content moderation on social media platforms, online hate speech, data breach notification laws and class action lawsuits,

cryptocurrencies, and cybercrime. The course requires no technical background beyond a willingness to learn a little about Internet architecture and software compromises and covers policy initiatives with an international perspective including case studies in the United States, European Union, Australia, Russia, China, Brazil, and India, among others.

CS 183: Privacy in the Digital Age

This course looks at issues of online privacy, including communications interception, encryption, online advertising, biometrics and facial recognition, and location tracking. No prior technical background is required, though the course will provide some background in encryption and network architecture to help students understand the mechanics of online surveillance and security efforts.

CS 184: Cyberlaw and Cyberpolicy

Wiretapping, copyright, and espionage haven't been the same since the Internet came into play—and law has been having a hard time keeping up with the changes that the new technology has wrought. This course is an introduction to the complex legal issues of cyberspace. Learn how the Digital Revolution impacts the First, Fourth, and Fifth Amendment law, the failure and solutions of copyright in the Digital Age, about the failures and limitations of the Computer Fraud and Abuse Act. The course will mostly focus on U.S. law and jurisprudence, but will cover issues from an international perspective. Plan to write, think, and argue!

Tufts Mobile