COMPUTER SCIENCE (CPSC)

* CPSC 035a / MUSI 035a, Twenty-First Century Electronic and Computer Music Techniques  Scott Petersen
Exploration of twenty-first century electronic and computer music through the diverse subjects and issues at the intersection of technology and new music. How computers have changed and challenged the analysis, composition, production, and appreciation of music over the last fifty years. Knowledge of basic music theory and the ability to read Western musical notation is assumed. Enrollment limited to first-year students. Preregistration required; see under First-Year Seminar Program.  QR

CPSC 100a / CPSC S100, Introduction to Computing and Programming  Benedict Brown and Cody Murphey
Introduction to the intellectual enterprises of computer science and to the art of programming. Students learn how to think algorithmically and solve problems efficiently. Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web development. Languages include C, Python, SQL, and JavaScript, plus CSS and HTML. Problem sets inspired by real-world domains of biology, cryptography, finance, forensics, and gaming. See CS50’s website, https://cs50.yale.edu, for additional information. No previous programming experience required. Open to students of all levels and majors.  QR

CPSC 112b, Introduction to Programming  Y. Richard Yang and Benedict Brown
Development on the computer of programming skills, problem-solving methods, and selected applications. No previous experience with computers necessary.  QR

CPSC 123b / PLSC 351b / S&D S 123b / S&D S 323b, YData: An Introduction to Data Science  John Lafferty and Elena Khushainova
Computational, programming, and statistical skills are no longer optional in our increasingly data-driven world; these skills are essential for opening doors to manifold research and career opportunities. This course aims to dramatically enhance knowledge and capabilities in fundamental ideas and skills in data science, especially computational and programming skills along with inferential thinking. YData is an introduction to Data Science that emphasizes the development of these skills while providing opportunities for hands-on experience and practice. YData is accessible to students with little or no background in computing, programming, or statistics, but is also engaging for more technically oriented students through extensive use of examples and hands-on data analysis. Python 3, a popular and widely used computing language, is the language used in this course. The computing materials will be hosted on a special purpose web server.  QR

* CPSC 150a, Computer Science and the Modern Intellectual Agenda  David Gelernter
Introduction to the basic ideas of computer science (computability, algorithm, virtual machine, symbol processing system), and of several ongoing relationships between computer science and other fields, particularly philosophy of mind. No previous experience with computers necessary. Enrollment limited to 25.  WR, HU

CPSC 183a, Law, Technology, and Culture  Brad Rosen
An exploration of the myriad ways in which law and technology intersect, with a special focus on the role of cyberspace. Topics include digital copyright, free speech, privacy and anonymity, information security, innovation, online communities, the impact of technology on society, and emerging trends. No previous experience with computers or law necessary.  SO

* CPSC 184b, Intellectual Property in the Digital Age  Cecillia Xie
The evolving and oftentimes vexing intellectual property regime of the new digital age. Focus on copyright, fair use, remix culture, access to knowledge, technological innovations, the increasing relevance of trademarks in the new information society, the tension between creativity/creating and the intellectual property rules which either foster or inhibit it, and the new information culture of the digital age. Prerequisite: CPSC 183 or permission of instructor.  HU, SO

* CPSC 185b, Control, Privacy, and Technology  Brad Rosen
The evolution of various legal doctrines with and around technological development. Topics include criminal law, privacy, search and seizure, digital rights, and the implications of technologically permitted methods of control on the law. Special attention to case law and policy. After CPSC 183.  WR, SO

CPSC 200a or b, Introduction to Information Systems  Stephen Slade
The real-world artifacts and implementations that comprise the vital computational organisms that populate our world. Hardware and software and the related issues of security, privacy, regulation, and software engineering. Examples stress practical applications of technology, as well as limitations and societal issues. After CPSC 100 or 112 or equivalent.  QR

CPSC 210a or b, Introduction to Computer Science  Staff
Introduction to the concepts, techniques, and applications of computer science. Topics include computer systems (the design of computers and their languages); theoretical foundations of computing (computability, complexity, algorithm design); and artificial intelligence (the organization of knowledge and its representation for efficient search). Examples stress the importance of different problem-solving methods. After CPSC 112 or equivalent.  QR

CPSC 220a or b, Mathematical Tools for Computer Science  Staff
Introduction to formal methods for reasoning and to mathematical techniques basic to computer science. Topics include propositional logic, discrete mathematics, and linear algebra. Emphasis on applications to computer science: recurrences, sorting, graph traversal, Gaussian elimination.  QR
CPSC 223a or b, Data Structures and Programming Techniques  
Staff  
Topics include programming in C; data structures (arrays, stacks, queues, lists, trees, heaps, graphs); sorting and searching; storage allocation and management; data abstraction; programming style; testing and debugging; writing efficient programs. After CPSC 201 or equivalent. QR RP

CPSC 257a, Information Security in the Real World  
Stephen Slade  
Introduction to information security, the practice of protecting information from unauthorized actions, in the context of computer systems. Topics include current security-related issues, basic adversarial models and threats to computer systems, potential defenses, security tools, and common security breaches and their wider impacts. Prerequisite: CPSC 100, 112, or equivalent programming experience, or with permission of instructor. QR

CPSC 276b, Introduction to Web Application for the Digital Humanities  
Benedict Brown  
Introduction to applications of computer and data science in the humanities, including web technologies, visualization, and database design. Students work in teams to develop a variety of applications proposed by faculty and staff from the Digital Humanities Lab, the Institute for the Preservation of Cultural Heritage, and the Computer Science department. Meets with CPSC 276. Students may earn credit for CPSC 276 or 376; not both. Prerequisite: CPSC 110, CPSC 112, equivalent programming experience, or permission of the instructor. QR

* CPSC 280a or b, Directed Reading  
James Aspnes  
Individual study for qualified students who wish to investigate an area of computer science not covered in regular courses. A student must be sponsored by a faculty member who sets the requirements and meets regularly with the student. Requires a written plan of study approved by the faculty adviser and the director of undergraduate studies. May be taken more than once for credit.

* CPSC 290a or b, Directed Research  
James Aspnes  
Individual research. Requires a faculty supervisor and the permission of the director of undergraduate studies. May be taken more than once for credit.

CPSC 323a or b, Introduction to Systems Programming and Computer Organization  
Staff  
Machine architecture and computer organization, systems programming in a high-level language, issues in operating systems, software engineering, prototyping in scripting languages. After CPSC 223. QR RP

CPSC 327a or b, Object-Oriented Programming  
Staff  
Object-oriented programming as a means to efficient, reliable, modular, reusable code. Use of classes, derivation, templates, name-hiding, exceptions, polymorphic functions, and other features of C++. This course was previously number CPSC 427. After CPSC 223. QR

CPSC 334b, Creative Embedded Systems  
Scott Petersen  
Ubiquitous computing is creating new canvases and opportunities for creative ideas. This class explores the use of microprocessors, distributed sensor networks, IoT, and intermedia systems for the purposes of creative expression. The course is delivered in a mixed lecture and lab format that introduces the fundamental concepts and theory behind embedded systems as well as issues particular to their creative employment. The key objective of the course is for students to perceive of and implement creative uses of computation. To this end, skills to be obtained during the course are as follows: (1) appreciate the current efforts and motivation to push the limitations of computation for creative expression, both in new application and new foundational research; (2) weigh factors such as cost, power, processing, memory, I/O capabilities, and networking capabilities when choosing a set of embedded devices and sensors; (3) contextualize unfamiliar hardware and languages through examples, documentation, and familiar design pattern; and (4) manage communication between multiple languages, devices, and protocols. Additionally, at the end of the course students will have a portfolio of their work in the form of writing, code, video, audio, and physical artifacts. Prerequisite: CPSC 223 or equivalent or by permission of instructor. QR RP

CPSC 335b, Theory and Implementation of Self-Driving Cars  
Man-Ki Yoon  
This course explores the theory and practice of building self-driving cars using advanced computing technologies. It aims to provide students opportunities i) to understand the introductory theory that enables the autonomous driving and also ii) to have extensive hands-on experience with various software and hardware tools. Topics include embedded system programming, sensor fusion, control theory, and introductory perception, planning and navigation techniques using machine learning and computer vision. Over the course of the semester, students work in small groups to design and build miniaturized self-driving cars that autonomously navigate an indoor track that resembles real road environments. Students demonstrate their learned skills through the final driving showcase and semester-long group projects. Meets with CPSC 235. Students may earn credit for CPSC 235 or for CPSC 335; not for both. Prerequisite: CPSC 223 and 202. Basic knowledge of Python is required. Instructor’s permission is required to waive the prerequisites. RP

CPSC 338b / EENG 348b, Digital Systems  
Rajit Manohar  
Development of engineering skills through the design and analysis of digital logic components and circuits. Introduction to gate-level circuit design, beginning with single gates and building up to complex systems. Hands-on experience with circuit design using computer-aided design tools and microcontroller programming. Recommended preparation: EENG 201. QR

CPSC 365b / ECON 365b, Algorithms  
James Glenn  
Paradigms for algorithmic problem solving: greedy algorithms, divide and conquer, dynamic programming, and network flow. NP completeness and approximation algorithms for NP-complete problems. Algorithms for problems from economics, scheduling, network
design and navigation, geometry, biology, and optimization. Provides algorithmic background essential to further study of computer science. Either CPSC 365 or CPSC 366 may be taken for credit. Prerequisites: CPSC 202 and 223. QR

* CPSC 366b / ECON 366b, Intensive Algorithms Yang Cai
Mathematically sophisticated treatment of the design and analysis of algorithms and the theory of NP completeness. Algorithmic paradigms including greedy algorithms, divide and conquer, dynamic programming, network flow, approximation algorithms, and randomized algorithms. Problems drawn from the social sciences, Data Science, Computer Science, and engineering. For students with a flair for proofs and problem solving. Either CPSC 365 or CPSC 366 may be taken for credit. Prerequisites: MATH 244 and CPSC 223. QR

CPSC 376b, Advanced Web Application Development in the Digital Humanities Benedict Brown
Advanced applications of computer and data science in the humanities, including web technologies, visualization, and database design. Students work in teams to develop a variety of applications proposed by faculty and staff from the Digital Humanities Lab, the Institute for the Preservation of Cultural Heritage, and the Computer Science department. Meets with CPSC 376. Students may earn credit for CPSC 276 or 376; not both. Prerequisite: CPSC 223 or equivalent, or permission of the instructor. QR

* CPSC 421b, Compilers and Interpreters Robert Soule
Compiler organization and implementation: lexical analysis, formal syntax specification, parsing techniques, execution environment, storage management, code generation and optimization, procedure linkage and address binding. The effect of language-design decisions on compiler construction. After CPSC 333. QR

CPSC 422a, Design and Implementation of Operating Systems Zhong Shao
The design and implementation of operating systems. Topics include synchronization, deadlock, process management, storage management, file systems, security, protection, and networking. After CPSC 322. QR

CPSC 423a / MUSI 428a, Computer Music: Algorithmic and Heuristic Composition Scott Petersen
Study of the theoretical and practical fundamentals of computer-generated music, with a focus on high-level representations of music, algorithmic and heuristic composition, and programming languages for computer music generation. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language. Ability to read music is assumed. After CPSC 202 and 222. QR

CPSC 424b, Parallel Programming Techniques Andrew Sherman
Practical introduction to parallel programming, emphasizing techniques and algorithms suitable for scientific and engineering computations. Aspects of processor and machine architecture. Techniques such as multithreading, message passing, and data parallel computing using graphics processing units. Performance measurement, tuning, and debugging of parallel programs. Parallel file systems and I/O. Prerequisite: CPSC 333, or CPSC 223 and significant experience with C/C++ programming in another science, social science or engineering discipline, or permission of instructor. QR RP

CPSC 425b, Cloud Networking and Systems
Study of critical technology trends and new challenges in cloud and data center designs for different trade-offs of performance, scalability, manageability, and cost in the networking layers and big data analytical frameworks. Consideration of cloud infrastructure, including network topology, network traffic management, network management, transport protocols, programmable switches, network functions, virtualization, network reliability, and security. After CPSC 433 or with permission of instructor. QR

CPSC 426a, Building an Internet Router Robert Soule
Over the course of the semester, students build a fully functioning Internet router. Students design the control plane in Python on a Linux host and design the data plane in the new P4 language on the bm2v2 software switch. To provide context and background for
the design of their router, students read a selection of papers to get both a historical perspective and exposure to current research in networking. Prerequisite: CPSC 433. SC

**CPSC 437a, Introduction to Database Systems**  
Avi Silberschatz  

**CPSC 439b, Software Engineering**  
Staff  
Introduction to fundamental concepts in software engineering and to the development and maintenance of large, robust software systems. The process of collecting requirements and writing specifications; project planning and system design; methods for increasing software reliability, including delta debugging and automatic test-case generation; type systems, static analysis, and model checking. Students build software in teams. After CPSC 323. QR

**CPSC 446a, Data and Information Visualization**  
Holly Rushmeier and Benedict Brown  
Visualization is a powerful tool for understanding data and concepts. This course provides an introduction to the concepts needed to build new visualization systems, rather than to use existing visualization software. Major topics are abstracting visualization tasks, using visual channels, spatial arrangements of data, navigation in visualization systems, using multiple views, and filtering and aggregating data. Case studies to be considered include a wide range of visualization types and applications in humanities, engineering, science, and social science. Prerequisite: CPSC 223.

**CPSC 452b, Deep Learning Theory and Applications**  
Smita Krishnaswamy  
Deep neural networks have gained immense popularity within the last decade due to their success in many important machine learning tasks such as image recognition, speech recognition, and natural language processing. This course provides a principled and hands-on approach to deep learning with neural networks. Students master the principles and practices underlying neural networks including modern methods of deep learning, and apply deep learning methods to real-world problems including image recognition, natural language processing, and biomedical applications. The course is based on homework, a final exam, and a final project (either group or individual, depending on the total number enrolled). The project includes both a written and oral (i.e. presentation) component. The course assumes basic prior knowledge in linear algebra and probability. Prerequisites: CPSC 202 and knowledge of Python Programming.

**CPSC 453a, Unsupervised Learning for Big Data**  
Smita Krishnaswamy  
This course focuses on machine-learning methods well-suited to tackling problems associated with analyzing high-dimensional, high-throughput noisy data including: manifold learning, graph signal processing, nonlinear dimensionality reduction, clustering and information theory. Though the class goes over some biomedical applications, such methods can be applied in any field. Prerequisite: Knowledge of linear algebra and Python Programming.

**CPSC 454a, Software Analysis and Verification**  
Ruzica Piskac  
Introduction to concepts, tools, and techniques used in the formal verification of software. State-of-the-art tools used for program verification; detailed insights into algorithms and paradigms on which those tools are based, including model checking, abstract interpretation, decision procedures, and SMT solvers. After CPSC 202 and 323 or equivalents. QR

**CPSC 455a / ECON 425a, Economics and Computation**  
Yang Cai  
A mathematically rigorous investigation of the interplay of economic theory and computer science, with an emphasis on the relationship of incentive-compatibility and algorithmic efficiency. Our main focus is on algorithmic tools in mechanism design, algorithms and complexity theory for learning and computing Nash and market equilibria, and the price of anarchy. Case studies in Web search auctions, wireless spectrum auctions, matching markets, and network routing, and social networks. Prerequisite: CPSC 365 or permission of the instructor. Familiarity with basic microeconomic theory is helpful but not required. QR

* **CPSC 456b / EENG 451b, Wireless Technologies and the Internet of Things**  
Wenjun Hu  
Over the last two decades or so, consumer IoT technologies have evolved from individual analogous devices, to connected devices and then interconnected networks of devices, from data collection to data management, from smart devices to intelligent interfaces. Wireless connectivity is an important driver of IoT technologies. This course aims to weave together fundamental theory of wireless communications, its application to IoT, and the design and implementation of wireless network architectures. The concepts are illustrated using examples such as WiFi and LTE/5G. Particular emphasis is placed on the interplay between concepts and their implementation in real systems. The coursework offers a practical experience, built on lab sessions involving WiFi experiments and simple IoT setups, homework involving Matlab-based analysis, and a student-defined course project that can cater to diverse interests. Students can expect to learn background knowledge of some everyday wireless technologies and how to design systems based on the fundamental communications concepts. Given the nature of these invisible signals, students also gain some experience of dealing with uncertainty in experiments and working towards open-ended goals. Depending on the programming background of the students, we may also explore backend system support in the form of edge or cloud computing. Prerequisites: 1) Introductory courses in mathematics, engineering, or computer science covering basics of the following topics: Linux skills, Matlab programming, probability, linear algebra, and Fourier transform; 2) Or by permission of the instructor. Course material will be self-contained as much as possible. The labs and homework assignments require Linux and Matlab skills and simple statistical and matrix analysis (using built-in Matlab functions). There will be a couple of introductory labs to refresh Linux and Matlab skills if needed.
* CPSC 457b, Sensitive Information in a Connected World  Michael Fischer
Issues of ownership, control, privacy, and accuracy of the huge amount of sensitive information about people and organizations that is collected, stored, and used by today's ubiquitous information systems. Readings consist of research papers that explore both the power and the limitations of existing privacy-enhancing technologies such as encryption and "trusted platforms." After or concurrently with CPSC 365 and 467.  QR

* CPSC 459a, Building Interactive Machines  Marynel Vazquez
This advanced course brings together methods from machine learning, computer vision, robotics, and human-computer interaction to enable interactive machines to perceive and act in a variety of environments. Part of the course examines approaches for perception with different sensing devices and algorithms; the other part focuses on methods for decision making and applied machine learning for control. Understanding of probability, differential calculus, linear algebra, and planning (in Artificial Intelligence) is expected for this course. Programming assignments require proficiency in Python and high-level familiarity with C++. Prerequisites: CPSC 201, CPSC 202, and CPSC 470 (or 570), or permission of the instructor.  QR

CPSC 460a, Automata Theory and Formal Languages  Andrew Bridy
Introduction to the theory of automata and formal languages, one of the building blocks of theoretical computer science. Major topics covered are finite automata, pushdown automata, and Turing machines, and their associated languages. Prerequisites: CPSC 201 (or equivalent) and CPSC 365/366/MATH 244 (or equivalent), or permission of instructor. Students should have some familiarity with formal mathematical argument, including proof techniques such as proof by induction and proof by contradiction.  QR

* CPSC 464b, Topics in Foundations of Machine Learning  Nisheeth Vishnoi
This course focuses on current and important topics in machine learning where a foundational understanding is lacking or under development. This includes modern algorithmic methods, novel learning and generative models, and the societal impact of machine learning. Representative topics include optimization and sampling methods for non-convex functions in Euclidean spaces or manifolds, algorithms beyond worst case, fairness, and robustness. This course is for students who would like to address the limitations of current machine learning systems deployed in the real world through a combination of foundational work such as coming up with the right definitions, modeling, methods, along with empirical evaluation. Prerequisites: CPSC 365 or 366 is required and S&D 252 is recommended. Solid background in calculus, linear algebra, stochastic processes, and advanced algorithms along with a good background in programming is necessary.

CPSC 465a, Theory of Distributed Systems  James Aspnes
Models of asynchronous distributed computing systems. Fundamental concepts of concurrency and synchronization, communication, reliability, topological and geometric constraints, time and space complexity, and distributed algorithms. After CPSC 365 or 366.  QR

CPSC 467a / CPSC 367, Cryptography and Computer Security  Michael Fischer
A survey of such private and public key cryptographic techniques as DES, RSA, and zero-knowledge proofs, and their application to problems of maintaining privacy and security in computer networks. Focus on technology, with consideration of such societal issues as balancing individual privacy concerns against the needs of law enforcement, vulnerability of societal institutions to electronic attack, export regulations and international competitiveness, and development of secure information systems. Some programming may be required. After CPSC 202 and 223.  QR

CPSC 469b, Randomized Algorithms  James Aspnes
A study of randomized algorithms from several areas: graph algorithms, algorithms in algebra, approximate counting, probabilistically checkable proofs, and matrix algorithms. Topics include an introduction to tools from probability theory, including some inequalities such as Chernoff bounds. After CPSC 365 or 366; a solid background in probability is desirable.  QR

CPSC 470b, Artificial Intelligence  Brian Scassellati
Introduction to artificial intelligence research, focusing on reasoning and perception. Topics include knowledge representation, predicate calculus, temporal reasoning, vision, robotics, planning, and learning. After CPSC 201 and 202.  QR

CPSC 474a, Computational Intelligence for Games  James Glenn
Introduction to techniques used for creating computer players for games, particularly board games. Topics include combinatorial and classical game theory, stochastic search methods, applications of neural networks, and procedural content generation. Prerequisites: CPSC 202 and CPSC 223.  QR

CPSC 475a / BENG 475a / EENG 475a, Computational Vision and Biological Perception  Steven Zucker
An overview of computational vision with a biological emphasis. Suitable as an introduction to biological perception for computer science and engineering students, as well as an introduction to computational vision for mathematics, psychology, and physiology students. Prerequisite: CPSC 112 and MATH 120, or with permission of instructor.  QR, SC  RP

CPSC 476b / BENG 476b, Advanced Computational Vision  Steven Zucker
Advanced view of vision from a mathematical, computational, and neurophysiological perspective. Emphasis on differential geometry, machine learning, visual psychophysics, and advanced neurophysiology. Topics include perceptual organization, shading, color and texture analysis, and shape description and representation. After CPSC 475.  QR, SC

CPSC 477b, Natural Language Processing  Dragomir Radev
Linguistic, mathematical, and computational fundamentals of natural language processing (NLP). Topics include part of speech tagging, Hidden Markov models, syntax and parsing, lexical semantics, compositional semantics, machine translation, text classification,
discourse, and dialogue processing. Additional topics such as sentiment analysis, text generation, and deep learning for NLP.
Prerequisites: CPSC 202 and CPSC 223, or permission of instructor.  

**CPSC 478b, Computer Graphics**  Theodore Kim
Introduction to the basic concepts of two- and three-dimensional computer graphics. Topics include affine and projective transformations, clipping and windowing, visual perception, scene modeling and animation, algorithms for visible surface determination, reflection models, illumination algorithms, and color theory. After CPSC 202 and 223.  

* **CPSC 479a, Advanced Topics in Computer Graphics**  Julie Dorsey
An in-depth study of advanced algorithms and systems for rendering, modeling, and animation in computer graphics. Topics vary and may include reflectance modeling, global illumination, subdivision surfaces, NURBS, physically-based fluids systems, and character animation. After CPSC 202 and 223.  

**CPSC 484b, Introduction to Human-Computer Interaction**  Marynel Vazquez
This course introduces students to the interdisciplinary field of Human-Computer Interaction (HCI), with particular focus on Human-Robot Interaction (HRI). The first part of the course covers principles and techniques in the design, development, and evaluation of interactive systems. It provides students with an introduction to UX Design and User-Centered Research. The second part focuses on the emergent field of HRI and several other non-traditional interfaces, e.g., AR/VR, tangibles, crowdsourcing. The course is organized as a series of lectures, presentations, a mid-term exam, and a semester-long group project on designing a new interactive system. After CPSC 201 and 202 or equivalents. Students who do not fit this profile may be allowed to enroll with the permission of the instructor.  

* **CPSC 490a or b, Senior Project**  James Aspnes
Individual research intended to fulfill the senior requirement. Requires a faculty supervisor and the permission of the director of undergraduate studies. The student must submit a written report about the results of the project.