**COMPUTER SCIENCE (CPSC)**

* CPSC 035b / MUSI 035b, 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  
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  
Timothy Barron

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 523b / S&D S 523b, YData: An Introduction to Data Science  
Ethan Meyers

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 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, 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 112 or equivalent.  
QR

* CPSC 201a or b, Introduction to Computer Science  
Stephen Slade and 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 202a 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 210b / PLSC 369b, Power, Security, and Surveillance: Political Challenges of the Computer Age  
Joan Feigenbaum

Twenty-first century societies are faced with both threats and opportunities that combine sophisticated computation with politics and international relations in critical ways. Examples include cyber warfare; cyber espionage; cyber crime; the role of social media in democratic self-governance, authoritarian control, and election "hacking"; cryptocurrencies; and mass surveillance. This course examines the political challenges wrought by massive increases in the power of computational and communication technologies and the potential for citizens and governments to harness those technologies to solve problems. It is co-taught by one faculty member in computer science
and one in political science. No previous programming experience required. Meets with CPSC 310. Students may earn credit for CPSC 210/PLSC 369 or for CPSC 310; not for both. Prerequisite: Internet literacy. SO

**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 262a / AMTH 262a / S&DS 262a, Computational Tools for Data Science**  Roy Lederman
Introduction to the core ideas and principles that arise in modern data analysis, bridging statistics and computer science and providing students the tools to grow and adapt as methods and techniques change. Topics include principal component analysis, independent component analysis, dictionary learning, neural networks and optimization, as well as scalable computing for large datasets. Assignments include implementation, data analysis and theory. Students require background in linear algebra, multivariable calculus, probability and programming. Prerequisites: after or concurrently with MATH 222, 225, or 231; after or concurrently with MATH 120, 230, or ENAS 151; after or concurrently with CPSC 100, 112, or ENAS 130; after S&D 100-108 or S&D 230 or S&D 241 or S&D 242. Enrollment is limited; requires permission of the instructor. QR

**CPSC 276b, Introduction to Web Application for the Digital Humanities**  Holly Rushmeier
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 376. 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, Directed Reading**  Staff
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, Directed Research**  Staff
Individual research. Requires a faculty supervisor and the permission of the director of undergraduate studies. May be taken more than once for credit.

**CPSC 310b, Technology, Power, and Security: Political Challenges of the Computer Age**  Joan Feigenbaum
Twenty-first century societies are faced with both threats and opportunities that combine sophisticated computation with politics and international relations in critical ways. Examples include cyber warfare; cyber espionage; cyber crime; the role of social media in democratic self-governance, authoritarian control, and election "hacking"; cryptocurrencies; and mass surveillance. This course examines the political challenges wrought by massive increases in the power of computational and communication technologies and the potential for citizens and governments to harness those technologies to solve problems. It is co-taught by one faculty member in computer science and one in political science. Programming experience and some knowledge of basic computer science is required. Meets with CPSC 210/PLSC 369. Students may earn credit for CPSC 210/PLSC 369 or for CPSC 310; not for both. Prerequisite: CPSC 223 or the equivalent. QR, SO

**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**  Timothy Barron
Object-oriented programming as a means to designing and writing efficient, reliable, modular, and reusable code. Covers core concepts and features of object-oriented languages (classes, inheritance, composition, encapsulation, polymorphism, and exceptions) as well as the use of object-oriented design patterns (iterator, decorator, strategy, adapter, observer, etc.). This course was previously number CPSC 427. After CPSC 223. QR

**CPSC 334a, 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 conceive 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 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 366b, Algorithms  Andre Wibisono
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  Dan Spielman
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 367b / CPSC 467b, Cryptography and Security  Staff
An introduction to cryptography and information security. Cryptographic algorithms and their application to security of digital data are presented. Some topics include classical, symmetric, and public key cryptography; digital signatures; cryptographic hash functions; and pseudorandom number generation. Multiparty protocols such as zero-knowledge proofs, secret sharing, anonymous communication, and secure multiparty function evaluation are introduced. Practical applications of cryptography to secure network communication, secure password authentication, and blockchains are also covered. The emphasis is on cryptographic algorithms and protocols that can be useful in providing information security. Students interested in a more mathematical and rigorous approach to these topics should take CPSC 467 instead, or in addition to this course. This course may not be taken for credit after CPSC 467. Prerequisites: Some programming required. After CPSC 202 and 223. QR

CPSC 376b, Advanced Web Application Development in the Digital Humanities  Holly Rushmeier
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 413a, Computer System Security  Timothy Barron
Overview of the principles and practice behind analyzing, designing, and implementing secure computer systems. Covers problems that have continued to plague computer systems for years as well as recent events and research in this rapidly evolving field of computer science. Learn to think from the perspective of an adversary; to understand systems well enough to see how their flaws could be exploited, and to consequently defend against such exploitation. Offers opportunities for hands-on exploration of attacks and defenses in the contexts of web applications, networks, and system level software. Also discusses ethical considerations and responsibilities associated with security research and practice. After CPSC 323.

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 323. QR

CPSC 425b, Mobile and Embedded Systems  Lin Zhong
Mobile and embedded systems are computers that are portable, embedded in a larger system, or both. They are usually resource constrained, intimately interact with the physical environment, including human users, and often serve mission-critical or privacy-sensitive applications. This course intends to provide a comprehensive introduction to the inner workings of modern mobile and embedded systems, from hardware architecture to operating systems to algorithms. While the lectures focus on theory, principle, and even historical lessons, significant learning of practical systems hacking skills, including learning itself, come from six programming assignments, involving Linux kernel development, FreeRTOS, and baremetal systems. Prerequisite: CPSC 323. SC

CPSC 429a, Principles of Computer System Design  Lin Zhong
Humans are stupid; computers are limited. Yet a collaboration of humans and computers has led to ever more powerful and complex computer systems. This course examines the limitations of humans and computers in this endeavor and how they shape the design, implementation, and evaluation of computer systems. It surveys the empirical knowledge reported by scholars and practitioners that overcome such limitations. The lectures, reading assignments, and classroom discussions travel through psychology and philosophy and revisit important results from theoretical computer science, with a goal of elucidating the rationales behind the best practices in computer systems research and development. Prerequisite: CPSC 323 or equivalent. Students should have the ability to write significant system programs in at least one systems programming language (e.g., C, C++ and Rust).

CPSC 431a / 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 223. QR

**CPSC 42b / MUSI 427b, Computer Music: Sound Representation and Synthesis**  Scott Petersen
Study of the theoretical and practical fundamentals of computer-generated music, with a focus on low-level sound representation, acoustics and sound synthesis, scales and tuning systems, 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 223. QR

**CPSC 43b, Computer Networks**  Anurag Khandelwal
An introduction to the design, implementation, analysis, and evaluation of computer networks and their protocols. Topics include layered network architectures, applications, transport, congestion, routing, data link protocols, local area networks, performance analysis, multimedia networking, network security, and network management. Emphasis on protocols used in the Internet. After CPSC 323. QR

* **CPSC 434a, Topics in Networked Systems**  Y. Richard Yang
Study of networked systems such as the Internet and mobile networks which provide the major infrastructure components of an information-based society. Topics include the design principles, implementation, and practical evaluation of such systems in new settings, including cloud computing, software-defined networking, 5G, Internet of things, and vehicular networking. Concurrently with or after CPSC 323. QR

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

* **CPSC 438a, Big Data Systems: Trends & Challenges**  Anurag Khandelwal
Today’s internet scale applications and cloud services generate massive amounts of data. At the same time, the availability of inexpensive storage has made it possible for these services and applications to collect and store every piece of data they generate, in the hopes of improving their services by analyzing the collected data. This introduces interesting new opportunities and challenges designing systems for collecting, analyzing and serving the so called “big data”. This course looks at technology trends that have paved the way for big data applications, survey state of the art systems for storage and processing of big data, and future research directions driven by open research problems. Our discussions span topics such as cluster architecture, big data analytics stacks, scheduling and resource management, batch and stream analytics, graph processing, ML/AI frameworks, serverless platforms and disaggregated architectures. Prerequisite: CPSC 323.

**CPSC 439b, Software Engineering**  Timos Antonopoulos
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 RP

**CPSC 446a, Data and Information Visualization**  Holly Rushmeier
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 448a / EENG 426a / ENAS 876a, Silicon Compilation**  Rajit Manohar
An upper-level course on compiling computations into digital circuits using asynchronous design techniques. Emphasis is placed on the synthesis of circuits that are robust to uncertainties in gate and wire delays by the process of program transformations. Topics include circuits as concurrent programs, delay-insensitive design techniques, synthesis of circuits from programs, timing analysis and performance optimization, pipelining, and case studies of complex asynchronous designs. Prerequisite: EENG 201 and introductory programming, or permission of instructor.

* **CPSC 449b / EENG 422b, Computer Architectures and Artificial Intelligence**  Richard Lethin
Introduction to the development of computer architectures specialized for cognitive processing, including both offline 'thinking machines' and embedded devices. The history of machines, from early conceptions in defense systems to contemporary initiatives. Instruction sets, memory systems, parallel processing, analog architectures, probabilistic architectures. Application and algorithm characteristics. Formerly EENG 449. Prerequisites: CPSC 100, CPSC 112, or equivalent programming experience; EENG 325, EENG 348, or equivalent circuits and digital logic experience; or permission of instructor. QR

* **CPSC 451b, The User Interface**  David Gelernter
The user interface (UI) in the context of modern design, where tech has been a strong and consistent influence from the Bauhaus and U.S. industrial design of the 1920s and 1930s through the IBM-Eames design project of the 1950s to 1970s. The UI in the context of the windows-menus-mouse desktop, as developed by Alan Kay and Xerox in the 1970s and refined by Apple in the early 1980s. Students develop a detailed design and simple implementation for a UI. Prerequisite: CPSC 223 or equivalent.
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 / NSCI 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 454b, Software Analysis and Verification  Staff
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 RP

* CPSC 456a / EENG 451a, 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 457a, 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." Junior or senior level standing with some background in computer science is required.

CPSC 458b, Automated Decision Systems  Stephen Slade
The spectrum of automated decision models and tools, with a focus on their costs and effectiveness. Examples from a variety of fields, including finance, risk management, robotics, medicine, and politics. After CPSC 223 or equivalents. QR

* CPSC 459a, Building Interactive Machines  Maryn Alvarez
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 463a, Algorithms via Continuous Optimization  Nisheeth Vishnoi
Continuous optimization has played a major role in the recent development of fast algorithms for problems arising in areas such as theoretical computer science, discrete optimization, and machine learning. The approach is to first formulate the problem as a continuous optimization problem, even if the problem may be over a discrete domain, adapt or develop deterministic or randomized continuous-
time dynamical systems to solve it, and then design algorithms for the problem via appropriate discretizations. The goal of this course is to design state-of-the-art algorithms for various classical discrete problems through the use of continuous optimization/sampling. The algorithmic applications include shortest paths, bipartite matching, flows, linear programming, sampling, and counting. We present approaches including gradient descent, mirror descent, multiplicative weights update method, accelerated gradient descent, Newton’s method, cutting plane methods, Langevin dynamics, and Hamiltonian dynamics. Prerequisites: CPSC 365 or 366 or permission of the instructor. S&DS 430 and a solid background in calculus, linear algebra, probability, and algorithms is recommended.  

CPSC 467b / CPSC 367b, Cryptography and Computer Security  Staff
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 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 472a, Intelligent Robotics  Brian Scassellati
Introduction to the construction of intelligent, autonomous systems. Sensory-motor coordination and task-based perception. Implementation techniques for behavior selection and arbitration, including behavior-based design, evolutionary design, dynamical systems, and hybrid deliberative-reactive systems. Situated learning and adaptive behavior. After CPSC 201 and 202 or equivalents. May not be taken after CPSC 473.  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, 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  Staff
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.  QR

CPSC 478b, Computer Graphics  Julie Dorsey
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.  QR

* 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 reflection modeling, global illumination, subdivision surfaces, NURBS, physically-based fluids systems, and character animation. After CPSC 202 and 223.  QR

CPSC 481a, Introduction to Machine Learning  Andre Wibisono
This course provides an introduction to machine learning and the problem of learning from data. It also introduces several frameworks for formulating the learning task as statistical and computational problems, and explores algorithms for solving them. Topics include supervised learning (classification, regression, kernel methods, neural networks), unsupervised learning (clustering, PCA, dimensionality reduction), reinforcement learning (Markov decision process, online learning), and examples of machine learning applications in various domains. This course provides a foundation for students interested in pursuing further research or applications of machine learning. Prerequisites: CPSC 202, MATH 222/225, S&DS 238, MATH 120, familiarity with programming.

* CPSC 490a, Senior Project  Staff
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.