Undergraduate Announcement

2024 - 2025

Computer Science

Program Offerings

A.B. Minor

With computation and computer science now permeating all corners of society and the economy, a computer science education has become a good launching pad for almost any career. Core concepts and skills emphasized in the computer science curriculum include theoretical and quantitative analysis of computation; design/engineering principles of advanced computer systems; and foundations and methods of AI and machine learning. The curriculum provides additional flexibility to explore subdisciplines of computer science (programming languages, formal methods, software engineering, computer graphics, information security), or to branch out into exciting cross-disciplinary investigations (neuroscience and cognitive science, computational biology, information policy, robotics, data science, etc.). Most computer science majors enjoy programming. Quite a few start with zero or minimal background and can enhance their skills while progressing through the curriculum.

The plan below applies to the Class of 2025 and beyond; the requirements for the Class of 2024 and earlier are available from the **Computer Science Department website** (https://www.cs.princeton.edu/ugrad/becoming-cs-major/requirements) and the archived version of the **Undergraduate Announcement** (https://ua.princeton.edu/archive).

Goals for Student Learning

The following are the key learning goals for a major in computer science:

- Students will be able to understand, modify, debug, refactor and analyze programs in a variety of different programming languages.
- Students will be able to learn new programming languages and computational paradigms they encounter in the future.
- Students will be able to read and comprehend technical literature in computer science.
- Students will be able to plan and complete a one- or two-term project in an area of computer science of their choice.
- Students will be able to identify, isolate and solve important technical subproblems that lead to a solution to a more significant problem.
- Students will be able to present technical ideas in computer science clearly, both orally and in written form.
- Students will be able to identify and assess related work in their area of study.
- Students will demonstrate an ability to work independently in accomplishing all of the above goals.

Prerequisites

The **prerequisites** (https://www.cs.princeton.edu/ugrad/becoming-cs-major/prerequisites) for A.B. students are any one of MAT 202/204/217 or EGR 154; COS 126 (or ECE 115); COS 217 and 226. Students should plan to take COS 126 in the first year. COS 217 and COS 226 **must** be completed *before* junior year. One or both of these are **prerequisites** (http://www.cs.princeton.edu/ugrad/becoming-cs-major/prerequisites) for all later computer science courses.

Admission to the Program

Information for First-Year Students. Students with a general interest in the sciences or engineering are encouraged to take **COS 126** (http://www.cs.princeton.edu/ugrad/becoming-cs-major/prerequisites) in the first year. This provides useful background for applications work in any science or engineering major and preserves the option of later electing a computer science major.

Program of Study

Departmental Requirements

Course Requirements: Majors must take at least **eight departmental courses** on a graded basis. These fall into three categories: foundation, core courses, and electives.

Foundation

Students must take COS 240 (Reasoning and Computation), to be completed before the end of junior year.

Core Courses

Students must take a total of four courses, one from each of the four categories listed below:

1. **Computer Systems**: COS 316 (Principles of Computer System Design) or COS 375 (Computer Architecture and Organization)

Alternatives:

- COS 318 (Operating Systems)
- COS 418 (Distributed Systems)
- COS 461 (Computer Networks)

2. **Artificial Intelligence and Machine Learning**: COS 324 (Introduction to Machine Learning)

Alternatives:

- COS 424 (Fundamentals of Machine Learning)
- COS 429 (Computer Vision)
- COS 484 (Natural Language Processing)

3. **Theoretical Computer Science**

- COS 423 (Theory of Algorithms)
- COS 433 (Cryptography)
- COS 445 (Networks, Economics, and Computing)
- COS 487 (Theory of Computation)

4. **Breadth**: This category contains courses that either explore another subdiscipline beyond Systems/Theory/AIML or provide experience with real-world applications. Students must complete at least one.

- COS 326 (Functional Programming)
- COS 333 (Advanced Programming Techniques)
- COS 343 (Algorithms for Computational Biology)
- COS 426 (Computer Graphics)
- COS 432 (Information Security)
- COS 436 (Human-Computer Interaction)
- COS 448 (Innovating across Technology, Business, and Markets)

Electives

Students must take three COS courses numbered at the **300 or 400 level** (500-level graduate courses do not count as electives). Alternatively, up to two electives may be chosen from a list of approved courses from other departments (see the **department website** (https://www.cs.princeton.edu/ugrad/class-of-2025-departmental-requirements) for an up-to-date list).

Students should consult with a computer science academic adviser on their course selections once they decide to become computer science majors. **Academic advisers** (http://www.cs.princeton.edu/ugrad/advisors) are listed on the Department of Computer Science webpage.

Interdisciplinary Studies

The pervasive nature of modern computing has introduced many interactions between computer science and other disciplines. Basic preparation in computer science is valuable for a broad variety of careers because of the computer's central role in society. Professionals who understand computers are far more effective in their work. In the past, a large amount of technical preparation was required before interesting applications could be considered; today's undergraduates can use computers to study important problems in other disciplines.

Some possible areas for interdisciplinary study are mathematics, music, art, economics, electrical and computer engineering, molecular biology, neurosciences and linguistics.

Many Princeton undergraduates view their four years at Princeton as an opportunity to gain an education before immersing themselves in rigorous training for careers in law, business or medicine. Computer science students are no exception. Through the choice of electives, students may create a specialized interdisciplinary program or a broad program with computer science as the core of preprofessional study. The former requires consultation with advisers in the related disciplines to determine what constitutes a reasonable cognate specialization, and the latter is constrained by the requirement of a coherent program of study in the major.

Independent Work

All A.B. majors will engage in **independent work** (http://www.cs.princeton.edu/ugrad/independent-work) supervised by a member of the department. IW projects involve the study and solution of specific problems in or related to computer science. These may arise from varied motivations, such as research questions intrinsic to the field; entrepreneurial activities; software design; policy or ethics issues in the tech world; applications of computer science to other disciplines or societal problems. Many students come up with their own IW topics; others may formulate them with help from faculty advisers.

In the fall term of junior year, majors must enroll in a **Junior Research Workshop (JRW)** (https://www.cs.princeton.edu/ugrad/independent-work/ab-junior-research-workshops), while simultaneously enrolling in a "Methods" 3xx course offered that term. The Methods course and the JRW complement each other and enable students to write a comprehensive proposal for a spring IW project by the end of the fall term, which can be used for a spring IW project. See the **department website** (https://www.cs.princeton.edu/ugrad/independent-work) for details.

A.B. junior majors then use the spring term to complete their proposed IW project — or, in many cases, embark on a separate research endeavor for the spring term — through one of the following advising structures:

1. One-on-One Advising. Students complete a project of their choosing while working one-on-one with a faculty adviser. Students may also work within the context of a faculty research project. Students can explore the **Undergraduate Research Topics** (https://www.cs.princeton.edu/ugrad/independent-work/undergraduate-research-topics) for faculty research interests and advisers.
2. **Independent Work Seminars** (https://www.cs.princeton.edu/ugrad/independent-work/independent-work-seminars). Students with shared interests around a common theme meet weekly as a group with a faculty adviser. Seminar instructors and themes vary term to term.

A.B. senior majors must complete a full-year senior thesis. The senior thesis may be a study in greater depth of one of the subjects considered in junior independent work, or it may deal with another aspect of computer science and its application. Students can explore the **Undergraduate Research Topics** (https://www.cs.princeton.edu/ugrad/independent-work/undergraduate-research-topics) for faculty research interests and advisers.

The **COS AB Independent Work and Thesis Handbook** (https://www.cs.princeton.edu/ugrad/independent-work) provides more information about the COS department's independent work and thesis requirements.

The department also offers a curriculum leading to a B.S.E. degree. The primary differences between the **A.B. and B.S.E.** (https://www.cs.princeton.edu/ugrad/becoming-cs-major/bse-vs-ab) programs are in the University requirements for the degree programs.

Senior Departmental Examination

An oral examination, consisting of a defense of the thesis research, will be held in April.

Study Abroad

Become a citizen of the world! **Computer science majors** (https://www.cs.princeton.edu/ugrad/study-abroad) are allowed and encouraged to study abroad for a semester or year. The **Office of International Programs** (http://www.princeton.edu/oip/home/) (OIP) manages study abroad programs. See the **Study Abroad Program** (http://www.princeton.edu/oip/sap/) page for specific information.

Faculty

Chair

Szymon M. Rusinkiewicz

Associate Chair

Ryan P. Adams
Wyatt A. Lloyd

Director of Undergraduate Studies

Brian W. Kernighan (co-director)
Kevin Wayne (co-director)

Director of Graduate Studies

Aarti Gupta (co-director)
Kyle A. Jamieson (co-director)

Professor

Ryan P. Adams
Andrew W. Appel
Sanjeev Arora
David I. August
Mark Braverman
Bernard Chazelle
Zeev Dvir
Adam Finkelstein
Michael J. Freedman
Tom Griffiths
Aarti Gupta
Elad Hazan
Kyle A. Jamieson
Brian W. Kernighan
Kai Li
Margaret R. Martonosi
Radhika Nagpal
Arvind Narayanan
Ben Raphael
Ran Raz
Jennifer Rexford
Szymon M. Rusinkiewicz
H. Sebastian Seung
Jaswinder P. Singh
Mona Singh
Robert E. Tarjan
Olga G. Troyanskaya
David P. Walker

Associate Professor

Jia Deng

Zachary Kincaid
Gillat Kol
Wyatt A. Lloyd
Jonathan Mayer
Karthik Narasimhan
Ravi A. Netravali
Olga Russakovsky
Matt Weinberg

Assistant Professor

Parastoo Abtahi
Danqi Chen
Tri Dao
Adji Bousso Dieng
Benjamin Eysenbach
Felix Heide
Peter Henderson
Aleksandra Korolova
Pravesh K. Kothari
Amit A. Levy
Lydia T. Liu
Alex Lombardi
Mae Milano
Andrés Monroy-Hernández
Yuri Pritykin
Huacheng Yu
Ellen Zhong

Associated Faculty

Amir Ali Ahmadi, Oper Res and Financial Eng
Christine Allen-Blanchette, Mechanical & Aerospace Eng
Maria Apostolaki, Electrical & Comp Engineering
Jianqing Fan, Oper Res and Financial Eng
Jaime Fernandez Fisac, Electrical & Comp Engineering
Yasaman Ghasempour, Electrical & Comp Engineering
Chi Jin, Electrical & Comp Engineering
Jason D. Lee, Electrical & Comp Engineering
Anirudha Majumdar, Mechanical & Aerospace Eng
Prateek Mittal, Electrical & Comp Engineering
Paul Seymour, Mathematics
John D. Storey, Integrative Genomics
Daniel L. Trueman, Music
Janet A. Vertesi, Sociology
Pramod Viswanath, Electrical & Comp Engineering
Mengdi Wang, Electrical & Comp Engineering
David Wentzlaff, Electrical & Comp Engineering

University Lecturer

Kevin Wayne

Senior Lecturer

Robert M. Dondero
Robert S. Fish
Alan Kaplan
Xiaoyan Li
Christopher M. Moretti

Lecturer

Sebastian Caldas
Marcel Dall'Agnol
Ruth C. Fong
Donna S. Gabai
Pedro Paredes
Iasonas Petras
Vikram V. Ramaswamy

For a full list of faculty members and fellows please visit the department or program website.

Courses

COS 109 - Computers in Our World (also EGR 109) Fall QCR

Computers are all around us. How does this affect the world we live in? This course is a broad introduction to computing technology for humanities and social science students. Topics will be drawn from current issues and events, and will include discussion of how computers work, what programming is and why it is hard, how the Internet and the Web work, security and privacy. Two 90-minute lectures. Self-scheduled computer laboratory. B. Kernighan

COS 126 - Computer Science: An Interdisciplinary Approach (also EGR 126) Fall/Spring QCR

An intro. to computer science in the context of scientific, engineering, and commercial applications. The goal of the course is to teach basic principles and practical issues, while at the same time preparing students to use computers effectively for applications in computer science, physics, biology, engineering, and other disciplines. Java programming language is used to introduce fundamental programming concepts, including conditionals, loops, arrays, functions, and object-oriented programming. Add topics include algorithms and data structures; theory of computing; and machine learning. Two lectures and two precepts. K. Wayne, A. Finkelstein

COS 217 - Introduction to Programming Systems Fall/Spring QCR

An introduction to computer organization and system software. Developing skills for composing large programs, including modularity, abstraction, programming style, and best practices for code development, testing, debugging, and performance tuning. Additionally, an overview of computing environments and architectures, through the C programming language, assembly language, and machine language. Two lectures and two precepts. C. Moretti, Z. Kincaid

COS 226 - Algorithms and Data Structures Fall/Spring QCR

This course surveys the most important algorithms and data structures in use on computers today. Topics include elementary data structures, sorting algorithms, search algorithms and data structures, graphs, and some more advanced topics (randomization, multiplicative weights and intractability). The course concentrates on developing implementations, understanding their performance characteristics, and estimating their potential effectiveness in applications. Two lectures and two precepts. G. Kol, K. Wayne

COS 240 - Reasoning About Computation Fall/Spring QCR

An introduction to mathematical topics relevant to computer science. Combinatorics, probability and graph theory will be covered in the context of computer science applications. The course will present a computer science approach to thinking and modeling. Students will be introduced to fundamental concepts in theoretical computer science, such as NP-completeness and cryptography that arise from the world view of efficient computation. R. Raz, I. Petras

COS 302 - Mathematics for Numerical Computing and Machine Learning (also ECE 305/SML 305) Fall

This course provides a comprehensive and practical background for students interested in continuous mathematics for computer science. The goal is to prepare students for higher-level subjects in artificial intelligence, machine learning, computer vision, natural language processing, graphics, and other topics that require numerical computation. This course is intended students who wish to pursue these more advanced topics, but who have not taken (or do not feel comfortable) with university-level multivariable calculus (e.g., MAT 201/203) and probability (e.g., ORF 245 or ORF 309). E. Zhong

COS 316 - Principles of Computer System Design Fall

This course teaches students the design, implementation, and evaluation of computer systems, including operating systems, networking, and distributed systems.The course will teach students to evaluate the performance and study the design choices of existing systems. Students will also learn general systems concepts that support design goals of modularity, performance, and security. Students will apply materials learned in lectures and readings to design and build new systems components. W. Lloyd, R. Fish

COS 320 - Compiling Techniques Not offered this year

The principal algorithms and concepts associated with translator systems. Topics include lexical analysis, syntactic analysis, parsing techniques, symbol table management, code generation and optimization, run time system design, implementation issues related to programming language design. Course will include a large-scale programming project utilizing the above topics. Three lectures. Prerequisites: 217 and 226 or instructor's permission. Staff

COS 324 - Introduction to Machine Learning Fall/Spring

This course is a broad introduction to different machine learning paradigms and algorithms and provides a foundation for further study or independent work in machine learning and data science. Topics include linear models for classification and regression, support vector machines, clustering, dimensionality reduction, deep neural networks, Markov decision processes, planning, and reinforcement learning. The goals of this course are three-fold: to understand the landscape of machine learning, how to compute the math behind techniques, and how to use Python and relevant libraries to implement and use various methods. J. Deng, R. Fong, L. Liu

COS 326 - Functional Programming Fall

An introduction to the principles of typed functional programming. Programming recursive functions over structured data types and informal reasoning by induction about the correctness of those functions. Functional algorithms and data structures. Principles of modular programming, type abstraction, representation invariants and representation independence. Parallel functional programming, algorithms and applications. A. Appel

COS 333 - Advanced Programming Techniques Fall/Spring

The practice of programming. Emphasis is on the development of real programs, writing code but also assessing tradeoffs, choosing among design alternatives, debugging and testing, and improving performance. Issues include compatibility, robustness, and reliability, while meeting specifications. Students will have the opportunity to develop skills in these areas by working on their own code and in group projects. Two lectures. Prerequisites: 217 and 226 (as corequisite). R. Dondero

COS 343 - Algorithms for Computational Biology Spring

This course introduces algorithms for analyzing DNA, RNA, and protein, the three fundamental molecules in the cell. Students will learn algorithms on strings, trees, and graphs and their applications in: sequence comparison and alignment; molecular evolution and comparative genomics; DNA sequencing and assembly; recognition of genes and regulatory elements; and RNA structure and protein interaction networks. Students will also implement algorithms and apply them to biological data. B. Raphael

COS 351 - Information Technology and Public Policy (also SOC 353/SPI 351) Not offered this year SA

This course surveys recurring, high-profile issues in technology policy and law. Each session will explore a challenging topic, including consumer privacy, data security, electronic surveillance, net neutrality, online speech, algorithmic fairness, cryptocurrencies, election security, and offensive operations. The seminar will also cover foundational technical concepts that affect policy and law, including internet architecture, cryptography, systems security, privacy science, and artificial intelligence. Materials and discussion will draw extensively from current events and primary sources. Staff

COS 375 - Computer Architecture and Organization (also ECE 375) Fall SEN

An introduction to computer architecture and organization. Instruction set design; basic processor implementation techniques; performance measurement; caches and virtual memory; pipelined processor design; design trade-offs among cost, performance, and complexity. Two classes Prerequisites: COS 217. M. Martonosi

COS 397 - Junior Independent Work (B.S.E. candidates only) Fall

Offered in the fall, juniors are provided with an opportunity to concentrate on a "state-of-the-art" project in computer science. Topics may be selected from suggestions by faculty members or proposed by the student. B.S.E. candidates only. R. Fish, D. Walker

COS 398 - Junior Independent Work (B.S.E. candidates only) Spring

Offered in the spring, juniors are provided with an opportunity to concentrate on a "state-of-the-art" project in computer science. Topics may be selected from suggestions by faculty members or proposed by the student. B.S.E. candidates only. R. Fish, A. Gupta

COS 417 - Operating Systems Spring

A study of the design and analysis of operating systems. Topics include: processes, mutual exclusion, synchronization, semaphores, monitors, deadlock prevention and detection, memory management, virtual memory, processor scheduling, disk management, file systems, security, protection, distributed systems. Two lectures. Prerequisites: 217 and 226 or instructor's permission. M. Milano, A. Levy

COS 418 - Distributed Systems Not offered this year

This course covers the design and implementation of distributed systems. Students will gain an understanding of the principles and techniques behind the design of modern, reliable, and high-performance distributed systems. Topics include server design, network programming, naming, concurrency and locking, consistency models and techniques, security, and fault tolerance. Modern techniques and systems employed at some of the largest Internet sites (e.g., Google, Facebook, Amazon) will also be covered. Through programming assignments, students will gain practical experience designing, implementing, and debugging real distributed systems. M. Freedman

COS 423 - Theory of Algorithms Spring

Design and analysis of efficient data structures and algorithms. General techniques for building and analyzing algorithms. Introduction to NP-completeness. Two lectures and one precept. Prerequisites: 226 and 240 or instructor's permission. R. Tarjan

COS 424 - Fundamentals of Machine Learning (also SML 302) Not offered this year

Computers have made it possible to collect vast amounts of data from a wide variety of sources. It is not always clear, however, how to use the data, and how to extract useful information from them. This problem is faced in a tremendous range of social, economic and scientific applications. The focus will be on some of the most useful approaches to the problem of analyzing large complex data sets, exploring both theoretical foundations and practical applications. Students will gain experience analyzing several types of data, including text, images, and biological data. Two 90-minute lectures. Prereq: MAT 202 and COS 126 or equivalent. Staff

COS 426 - Computer Graphics Fall

The principles underlying the generation and display of graphical pictures by computer. Hardware and software systems for graphics. Topics include: hidden surface and hidden line elimination, line drawing, shading, half-toning, user interfaces for graphical input, and graphic system organization. Two 90-minute lectures. Prerequisites: 217 and 226. A. Finkelstein

COS 429 - Computer Vision Fall

An introduction to the concepts of 2D and 3D computer vision. Topics include low-level image processing methods such as filtering and edge detection; segmentation and clustering; optical flow and tracking; shape reconstruction from stereo, motion, texture, and shading. Throughout the course, there will also be examination of aspects of human vision and perception that guide and inspire computer vision techniques. Prerequisites: 217 and 226. Two lectures. O. Russakovsky, V. Ramaswamy

COS 432 - Information Security (also ECE 432) Spring

Security issues in computing, communications, and electronic commerce. Goals and vulnerabilities; legal and ethical issues; basic cryptology; private and authenticated communication; electronic commerce; software security; viruses and other malicious code; operating system protection; trusted systems design; network security; firewalls; policy, administration and procedures; auditing; physical security; disaster recovery; reliability; content protection; privacy. Prerequisites: 217 and 226. Two lectures. Staff

COS 433 - Cryptography (also MAT 473) Fall

An introduction to the theory of modern cryptography. Topics covered include private key and public key encryption schemes, digital signatures, pseudorandom generators and functions, zero-knowledge proofs, and some advanced topics. Prerequisites: COS 240 is a required prerequisite or equivalent proof-based mathematical maturity. Two lectures. A. Lombardi

COS 436 - Human-Computer Interaction Fall

This course is a survey of the field of Human-Computer Interaction (HCI). We cover foundational as well as current research topics in HCI. The two core focus areas are interactive computing and social computing; breadth topics include AI+HCI, AR/VR, design tools, and accessibility. A semester-long group project that involves the design and implementation of an interactive system. Prerequisite COS 217 or permission of instructor. Two lectures and one precept. A. Monroy-Hernández

COS 445 - Economics and Computing Spring

Computation and other aspects of our lives are becoming increasingly intertwined. In this course we will study a variety of topics on the cusp between economics and computation. Topics to be covered include: games on networks, auctions, mechanism and market design, reputation, computational social choice. The aim of the course is two-fold: (1) to understand the game-theoretic issues behind systems involving computation such as online networks, and (2) to learn how algorithms and algorithmic thinking can help with designing better decision and allocation mechanisms in the offline world. M. Weinberg, M. Braverman

COS 448 - Innovating Across Technology, Business, and Marketplaces (also EGR 448) Spring

This course introduces computer science and technology-oriented students to issues tackled by Chief Technology Officers: the technical visionaries and managers innovating at the boundaries of technology and business. These individuals are partners to the business leaders of the organization, not merely implementers of business goals. The course covers companies from ideation

guest lecture. J. Singh, R. Fish

COS 451 - Computational Geometry Not offered this year

Introduction to basic concepts of geometric computing, illustrating the importance of this new field for computer graphics, solid modelling, robotics, databases, pattern recognition, and statistical analysis. Algorithms for geometric problems. Fundamental techniques, for example, convex hulls, Voronoi diagrams, intersection problems, multidimensional searching. Two 90-minute lectures. Prerequisites: 226 and 240 or 341, or equivalent. Staff

COS 461 - Computer Networks Spring

This course studies computer networks and the services built on top of them. Topics include packet-switch and multi-access networks, routing and flow control, congestion control and quality-of-service, Internet protocols (IP, TCP, BGP), the client-server model and RPC, elements of distributed systems (naming, security, caching) and the design of network services (multimedia, peer-to-peer networks, file and Web servers, content distribution networks). Two lectures, one preceptorial. Prerequisite: 217. K. Jamieson

COS 480 - Special Topics in Computer Science Not offered this year

These courses cover one or more advanced topics in computer science. The courses are offered only when there is an opportunity to present material not included in the established curriculum; the subjects vary from term to term. Three classes. Staff

COS 481 - Special Topics in Computer Science Not offered this year

These courses cover one or more advanced topics in computer science. The courses are offered only when there is an opportunity to present material not included in the established curriculum; the subjects vary from term to term. Three classes. Staff

COS 484 - Natural Language Processing Spring

Recent advances have ushered in exciting developments in natural language processing (NLP), resulting in systems that can translate text, answer questions and even hold spoken conversations with us. This course will introduce students to the basics of NLP, covering standard frameworks for dealing with natural language as well as algorithms and techniques to solve various NLP problems, including recent deep learning approaches. Topics covered include language modeling, rep. learning, text classification, sequence tagging, syntactic parsing, and machine translation. The course will have programming assignments, a mid-term and a final project. D. Chen, V. Ramaswamy

COS 487 - Theory of Computation (also MAT 407) Not offered this year

Studies the limits of computation by identifying tasks that are either inherently impossible to compute, or impossible to compute within the resources available. Introduces students to computability and decidability, Godel's incompleteness theorem, computational complexity, NP-completeness, and other notions of intractability. This course also surveys the status of the P

versus NP question. Additional topics may include: interactive proofs, hardness of computing approximate solutions, cryptography, and quantum computation. Two lectures, one precept. Prerequisite: 240 or 341, or instructor's permission. Staff

COS 488 - Introduction to Analytic Combinatorics (also MAT 474) Not offered this year

Analytic Combinatorics aims to enable precise quantitative predictions of the properties of large combinatorial structures. The theory has emerged over recent decades as essential both for the scientific analysis of algorithms in computer science and for the study of scientific models in many other disciplines. This course combines motivation for the study of the field with an introduction to underlying techniques, by covering as applications the analysis of numerous fundamental algorithms from computer science. The second half of the course introduces Analytic Combinatorics, starting from basic principles. Staff

COS 497 - Senior Independent Work (B.S.E. candidates only) Fall

Offered in the fall, seniors are provided with an opportunity to concentrate on a "state-of-the-art" project in computer science. Topics may be selected from suggestions by faculty members or proposed by the student. B.S.E. candidates only. R. Fish, D. Walker

COS 498 - Senior Independent Work (B.S.E. candidates only) Spring

Offered in the spring, seniors are provided with an opportunity to concentrate on a "state-of-the-art" project in computer science. Topics may be selected from suggestions by faculty members or proposed by the student. B.S.E. candidates only. R. Fish, A. Gupta

ECE 206 - Contemporary Logic Design (also COS 306) Fall SEL

Introduction of the basic concepts in logic design that form the basis of computation and communication circuits. This course will start from scratch and end with building a working computer on which we will run small programs. S. Malik, H. Valavi

ECE 381 - Networks: Friends, Money and Bytes (also COS 381) Not offered this year

This course is oriented around 20 practical questions in the social, economic, and technological networks in our daily lives. How does Google sell ad spaces and rank webpages? How does Netflix recommend movies and Amazon rank products? How do I influence people on Facebook and Twitter? Why doesn't the Internet collapse under congestion, and does it have an Achilles heel? Why does each gigabyte of mobile data cost $10, but Skype is free? How come Wi-Fi is slower at hotspots than at home, and what is inside the cloud of iCloud? In formulating and addressing these questions, we introduce the fundamental concepts behind the networking industry. Staff

ECE 396 - Introduction to Quantum Computing (also COS 396) Fall

This course will introduce the matrix form of quantum mechanics and discuss the concepts underlying the theory of quantum information. Some of the important algorithms will be discussed, as well as physical systems which have been suggested for quantum computing. Prerequisite: Linear algebra at the level of MAT 202, 204, 217, or the equivalent. S. Gopalakrishnan

ECE 462 - Design of Very Large-Scale Integrated (VLSI) Systems (also COS 462) Fall

Analysis and design of digital integrated circuits using deep sub-micron CMOS technologies as well as emerging and post-CMOS technologies (Si finFETs, III-V, carbon). Emphasis on design, including synthesis, simulation, layout and post-layout verification. Analysis of energy, power, performance, area of logic-gates, interconnect and signaling structures. N. Verma

ECE 475 - Computer Architecture (also COS 475) Spring

An in-depth study of the fundamentals of modern computer processor and system architecture. Students will develop a strong theoretical and practical understanding of modern, cutting-edge computer architectures and implementations. Studied topics include: Instruction-set architecture and high-performance processor organization including pipelining, out-of-order execution, as well as data and instruction parallelism. Cache, memory, and storage architectures. Multiprocessors and multicore processors. Coherent caches. Interconnection and network infrastructures. Prerequisite: ECE 375/COS 375 and ECE 206/COS 306 (or familiarity with Verilog). D. Wentzlaff

MAE 345 - Introduction to Robotics (also COS 346/ECE 345/ROB 345) Fall

Robotics is a rapidly-growing field with applications including unmanned aerial vehicles, autonomous cars, and robotic manipulators. This course will provide an introduction to the basic theoretical and algorithmic principles behind robotic systems.

The course will also allow students to get hands-on experience through project-based assignments. Topics include inverse kinematics, motion planning, localization, mapping, vision, and reinforcement learning. Prerequisites: MAT 201 or 203, MAT 202 or 204, COS 126. Recommended ORF 309 and MAE 305. A.B. students ST requirement; B.S.E. students 1st-year science requirement. Two 90-minute lectures. R. Nagpal

MAT 375 - Introduction to Graph Theory (also COS 342) Spring QCR

The fundamental theorems and algorithms of graph theory. Topics include: connectivity, matchings, graph coloring, planarity, the four-color theorem, extremal problems, network flows, and related algorithms. Prerequisite: MAT202 or 204 or 217 or equivalent. Staff

MUS 314 - Computer and Electronic Music through Programming, Performance, and Composition (also COS 314) Fall QCR

An introduction to the fundamentals of computer and electronic music in the context of the Princeton Laptop Orchestra (PLOrk). The music and sound programming language ChucK, developed here at Princeton, will be used in conjunction with Max/MSP, another digital audio language, to study procedural programming, digital signal processing and synthesis, networking, and human-computer interfacing. D. Trueman, J. Snyder

ORF 363 - Computing and Optimization for the Physical and Social Sciences (also COS 323) Fall/Spring QCR

An introduction to several fundamental and practically-relevant areas of modern optimization and numerical computing. Topics include computational linear algebra, first and second order descent methods, convex sets and functions, basics of linear and semidefinite programming, optimization for statistical regression and classification, and techniques for dealing with uncertainty and intractability in optimization problems. Extensive hands-on experience with high-level optimization software. Applications drawn from operations research, statistics and machine learning, economics, control theory, and engineering. A. Ahmadi

QCB 455 - Introduction to Genomics and Computational Molecular Biology (also COS 455/MOL 455) Fall QCR

This interdisciplinary course provides a broad overview of computational and experimental approaches to decipher genomes and characterize molecular systems. We focus on methods for analyzing "omics" data, such as genome and protein sequences, gene expression, proteomics and molecular interaction networks. We cover algorithms used in computational biology, key statistical concepts (e.g., basic probability distributions, significance testing, multiple testing correction, performance evaluation), and machine learning methods which have been applied to biological problems (e.g., classification techniques, hidden Markov models, clustering). J. Akey, M. Singh

---