

COMPUTER SCIENCE (COMP)

COMP112 Introduction to Programming

The course will provide an introduction to a modern, high-level programming language including a discussion of input/output, basic control structures, types, functions, and classes. The lectures will also discuss a variety of algorithms as well as program design issues.

The second meeting time for each section is a computer lab.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **None**

COMP112Z Introduction to Programming

This course will provide an introduction to imperative programming including a discussion of variables, basic control structures, and functions, with other topics as time permits. The lectures will also discuss a variety of algorithms as well as program design issues. The language is typically a standard high-level language like Java or Python, but the choice is up to the instructor, and may differ between different COMP 112 sections.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **None**

COMP113 Bioinformatics Programming

This course is an introduction to bioinformatics and programming for students with interest in the life sciences. It introduces problem areas and conceptual frameworks in bioinformatics. The course assumes little or no prior programming experience and will introduce the fundamental concepts and mechanisms of computer programs and examples (e.g., sequence matching and manipulation, database access, output parsing, dynamic programming) frequently encountered in the field of bioinformatics.

Offering: **Crosslisting**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-BIOL**

Identical With: **BIOL265, MB&B265, CIS265**

Prereq: **[MB&B181 or BIOL181]**

COMP114 How to Talk to Machines

How do we tell robots and computers how to do what they do? Getting a handle on this question is the goal of this course. Since telling a device how to do something depends a lot on what that device can do, along the way we will learn a bit about what is "in the box." We will start with the kind of programming one might use to instruct a robot how to interact with the world around it. That will lead us to the Turing machine, a beautiful mathematical model of a computer. We will adapt that model to something that is closer to how most computer systems today are designed. We will end with an introduction to high-level programming, learning the fundamentals of programming in a language such as Python or Java.

The goal of the course is to understand not just programming, but how computers are designed, and how those designs are reflected in the way we program them. After passing this course, students will have a basic knowledge of

programming and how a computer works. COMP 114 can be used to satisfy the COMP 211 prerequisite and also the mathematics major "elementary knowledge of algorithms and computer programming" requirement.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **None**

COMP114Z How to Talk to Machines

How do we tell robots and computers how to do what they do? Getting a handle on this question is the goal of this course. Since telling a device how to do something depends a lot on what that device can do, along the way we will learn a bit about what is "in the box." We will start with the kind of programming one might use to instruct a robot how to interact with the world around it. That will lead us to the Turing machine, a beautiful mathematical model of a computer. We will adapt that model to something that is closer to how most computer systems today are designed. We will end with an introduction to high-level programming, learning the fundamentals of programming in a language such as Python or Java.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **None**

COMP115 How to Design Programs

In this course, students will learn to systematically design programs, going from a problem statement to a well-organized solution in a step-by-step fashion. We will apply these program design skills to many applications within computer science and in other disciplines. Students will develop their mathematical skills, because we will use a symbolic view of computation that explains the process of running a program as simple manipulations of its text. Students will also develop their technical reading and writing skills, such as understanding complex problem descriptions and precisely articulating the design of solutions. No prior experience with programming or computer science is expected.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **None**

COMP131F Can Machines Think? (FYS)

This course is a freshman seminar (an FYS course) that introduces students to foundational ideas in logic, formal languages, and computation that have been the basis of a revolution in thinking in the 20th century. The material includes a substantial logic component and an introduction to programming in SML, a so-called functional programming language, as well as readings about the philosophical impact of these ideas.

The impact of the new cycle of ideas mentioned above has been felt far beyond the confines of the computer science community. Ideas from logic, computation, and other branches of mathematics formerly regarded as "pure" have spread to computation, linguistics, cognitive science, and philosophy; conversely, results from philosophy, linguistics, and other areas have been applied in computer science. New "non-traditional" logics have found unexpected applications in computing.

Traditionally, the background in logic and computation upon which these ideas rest has been taught in advanced courses, although much of it is accessible to students with little or no background in the area (but some willingness to learn formal reasoning and mathematical arguments). Students will learn about the underpinnings of recent technical advances in these areas at an unusually early stage in their careers. We will learn how to code basic programs in SML as well. The course will continue with readings in expository texts about the impact and philosophical significance of important ideas in logic and computation. Some readings in Philosophy of Mind will be assigned and discussed.

Offering: Host

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Prereq: None

COMP211 Computer Science I

This is the first course in a two-course sequence (COMP211-212) that is the gateway to the computer science major. It is intended for prospective computer science majors and others who want an in-depth understanding of programming and computer science. Topics to be covered in COMP211-212 include an introduction to the fundamental ideas of programming in imperative and functional languages, correctness and cost specifications, and proof techniques for verifying specifications.

Specifics such as choice of programming language, which topics are covered in which semesters, etc., will vary according to the tastes of the faculty offering the courses.

Offering: Host

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Prereq: COMP112 OR COMP113 OR COMP114 OR COMP115

COMP212 Computer Science II

This is the second course in a two-course sequence (COMP211-212) that is the gateway to the computer science major. It is intended for prospective computer science majors and others who want an in-depth understanding of programming and computer science. Topics to be covered in COMP211-212 include an introduction to the fundamental ideas of programming in imperative and functional languages, correctness and cost specifications, and proof techniques for verifying specifications.

Specifics such as choice of programming language, which topics are covered in which semesters, etc., will vary according to the tastes of the faculty offering the courses.

Offering: Host

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Prereq: COMP211

COMP260 Special Topics in Computer Science

This course is designed for nonmajors who wish to pursue some topic in computer science beyond introduction to programming. Topics will vary according to the instructor.

Offering: Host

Grading: A-F

Credits: 1.00

Gen Ed Area: NSM-MATH

Identical With: QAC260

Prereq: COMP112

COMP260A Software Design and Engineering

Programming is only one ingredient for creating industry-grade software. Eliciting requirements from stakeholders, modeling the architecture of a system, selecting appropriate development tools, and testing and maintaining a codebase are some of the aspects that elevate programming to software engineering. Focusing on the development of mobile and web apps, students in this course will gain expertise in common front-end and back-end technologies as well as related tooling. We will also cover the organization of software projects and their social implications, which are so important to recognize for the modern software engineer.

Offering: Host

Grading: A-F

Credits: 1.00

Gen Ed Area: None

Prereq: COMP112

COMP260B Scientific Computing

This course will introduce students from various disciplines to computing as applied to scientific problems. The course aims to provide a theoretical background in computational sciences and insight in understanding how popular algorithms and advances in hardware are used to solve problems drawn from chemistry, physics, biology, and bioinformatics. Topics covered will include simulation techniques, analysis of trajectories, clustering, probabilistic modeling, networks, artificial intelligence, neural networks, population dynamics, and high-performance computing.

Offering: Host

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Prereq: COMP211

COMP266 Bioinformatics

This course is an introduction to bioinformatics for students with interest in the life sciences. The course is similar to BIOL265 but only meets in the second half of the semester (with BIOL265) and is designed for students with programming background, ideally in Python. The course introduces problem areas and conceptual frameworks in bioinformatics and discusses programming approaches used in bioinformatics such as sequence matching and manipulation algorithms using dynamic programming, clustering analysis of gene expression data, analysis of genetic nets using Object Oriented Programming, and sequence analysis using Hidden Markov Models, Regular Expressions, and information theory.

Offering: Crosslisting

Grading: A-F

Credits: 0.50

Gen Ed Area: NSM-BIOL

Identical With: BIOL266, MB&B266, CIS266

Prereq: [MB&B181 OR BIOL181]

COMP301 Automata Theory and Formal Languages

This course is an introduction to formalisms studied in computer science and mathematical models of computing machines. The language formalisms discussed will include regular, context-free, recursive, and recursively enumerable languages. The machine models discussed include finite-state automata, pushdown automata, and Turing machines.

Offering: Host

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Identical With: COMP500

Prereq: COMP211 AND COMP212 AND MATH228

COMP312 Algorithms and Complexity

The course will cover the design and analysis of efficient algorithms. Basic topics will include greedy algorithms, divide-and-conquer algorithms, dynamic programming, and graph algorithms. Some advanced topics in algorithms may be selected from other areas of computer science.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Identical With: **COMP510**

Prereq: **COMP212 AND MATH228**

COMP321 Design of Programming Languages

This course is an introduction to concepts in programming languages. Topics include parameter passing, type checking and inference, control mechanisms, data abstraction, module systems, and concurrency. Basic ideas in functional, object-oriented, and logic programming languages will be discussed.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Identical With: **COMP521**

Prereq: **COMP212 AND MATH228**

COMP323 Programming Language Implementation

This course is an introduction to the implementation of programming languages.

Students will learn how to formally describe and implement major components of the implementation pipeline. Topics may include lexical analysis and parsing (checking whether source code is well-formed and converting it to an internal programmatic representation), type-checking and -inference (static program analysis for safety features), interpretation (direct execution of a high-level language program), and compilation (translation to a low-level language such as assembly or bytecode).

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Identical With: **COMP523**

Prereq: **COMP212 AND MATH228**

COMP327 Evolutionary and Ecological Bioinformatics

Bioinformatic analysis of gene sequences and gene expression patterns has added enormously to our understanding of ecology and evolution. For example, through bioinformatic analysis of gene sequences, we can now reconstruct the evolutionary history of physiology, even though no traces of physiology exist in the fossil record. We can determine the adaptive history of one gene and all the gene's descendants. We can now construct the evolutionary tree of all of life. Bioinformatics is particularly promising for analysis of the ecology and biodiversity of microbial communities, since well over 99 percent of microorganisms cannot be cultured; our only knowledge of these organisms is through analysis of their gene sequences and gene expression patterns. For example, even when we cannot culture most of a microbial community, we can determine which metabolic pathways are of greatest significance through analysis of community-level gene expression. All these research programs are made accessible not only by breakthroughs in molecular technology but also by innovation in the design of computer algorithms. This course, team-taught by an evolutionary biologist and a computer scientist, will present how bioinformatics is revolutionizing evolutionary and ecological investigation and will present the design and construction of bioinformatic computer algorithms underlying the revolution in biology. Students will learn algorithms for reconstructing phylogeny, for sequence alignment, and for analysis of genomes, and students will have an opportunity to create their own algorithms.

Offering: Crosslisting

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-BIOL**

Identical With: **BIOL327, BIOL527, COMP527, CIS327**

Prereq: **[BIOL182 or MB&B182] OR [BIOL196 or MBB196] OR COMP112 OR COMP211**

COMP331 Computer Structure and Organization

The purpose of the course is to introduce and discuss the structure and operation of digital computers. Topics will include the logic of circuits, microarchitectures, microprogramming, conventional machine architectures, and an introduction to software/hardware interface issues. Assembly language programming will be used to demonstrate some of the basic concepts.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Identical With: **COMP531**

Prereq: **COMP212**

COMP332 Computer Networks

This course will provide an introduction to the fundamentals of computer networks. Computer networks have become embedded in our everyday lives, from the Internet to cellular phones to cloud networking, enabling applications such as email, texting, web browsing, on-demand video, video conferencing, peer-to-peer file sharing, social networking, cloud computing, and more.

This course will delve into the infrastructure and protocols that have allowed computer networks to achieve their current ubiquity. While the primary focus of the course will be on the Internet's architecture, protocols, and applications, we will also touch on other types of computer networks. Programming assignments will be done using Python; prior knowledge of Python is not required.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 and MATH228**

COMP333 Software Engineering

Software engineering is the application of engineering principles to the software development process. Eliciting requirements from stakeholders, designing the architecture of a program, performing usability studies, and testing a codebase are some of the aspects that elevate program development to software engineering. Focusing on web and mobile apps, students in this course will gain expertise in state-of-the-art frontend, backend, and mobile technologies, as well as related tooling. We will also cover the collaborative organization of software projects, software licensing, software business models, and ethical considerations for professional software engineers.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP211 AND COMP212**

COMP341 Artificial Intelligence

This course is an introduction to creating programs that appear to behave intelligently. Topics will include search algorithms for problem solving, as well as probabilistic reasoning, including regression, classification, and decision making. Sample topics include Bayesian networks, basic neural networks and reinforcement learning.

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP343 Machine Learning

This course will provide an introduction to machine learning. The field of machine learning studies how to design systems that learn from experience. We will cover fundamental concepts and algorithms used in machine learning, as well as give an introduction to basic probability and statistics. Sample topics include regression, classification, Bayesian networks, Gibbs sampling, particle filtering, maximum likelihood estimation, neural networks, deep learning, clustering, bias/variance trade-offs, cross-validation, and practical advice. Programming assignments will be done using Python; prior knowledge of Python is not required.

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP350 Computational Media: Videogame Development

This course examines the interplay of art and science in the development of contemporary videogames using the Unity development platform and commercial artistic game tools. Students develop a comprehensive understanding of computational media, including legal and commercial aspects, combined with hands-on experience in a creative process that integrates design, art, and coding. There will be discussions with invited industry leaders in various subject areas. Students will have the opportunity to work as part of development teams and create working prototypes to better understand the challenges and rewards of producing graphic interactive software within a professional context.

Offering: **Crosslisting**

Grading: **A-F**

Credits: **2.00**

Gen Ed Area: **NSM-IDEA**

Identical With: **IDEA350, FILM250, CIS350**

Prereq: **None**

COMP353 Robotics

Students will use hands-on projects, building and programming Arduino-based robots to learn about the field of robotics. Topics will include perception, locomotion, localization, common programming frameworks for robotics, and ethical considerations. Some background in electronics and mechanical construction may be helpful but is not required.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP211**

COMP360 Special Topics in Computer Science: Information Security and Privacy

This course explores principles and practical applications of computer security and privacy. Some of the topics covered include static and dynamic code analysis, secure authentication, privacy enhancing technologies, usable privacy and security, and web tracking. We will also touch upon theoretical areas, such as basic cryptographic concepts as well as differential privacy. The course has the objective to provide students with the conceptual knowledge and technical skills to identify and resolve privacy and security issues in the design, development, and evaluation of information systems.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP360A Special Topics in Computer Science

The course will be an introduction to automated reasoning for computer science. Decision procedures for the satisfiability of formulas of propositional logic and

of various fragments of first-order equational theories will be covered. For the application part of the course, students will use the automated modeling tool Alloy to design and analyze software systems. Student work will include both mathematical exercises and programming exercises.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP360B Special Topics in Computer Science

This course covers special topics in computer science. Topics will vary according to the instructor.

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP360C Special Topics in Computer Science

Special Topics in Computer Science

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP360D Special Topics in Computer Science

Special Topics in Computer Science

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP360E Special Topics in Computer Science: Functional Data Structures

Advanced data structures from a functional programming perspective. The focus will be on the use of lazy evaluation as a technique for designing efficient data structures, especially in the presence of persistence (when the state of the structure must be preserved even after updates). Assessment will consist primarily of programming and written assignments.

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP360G Special Topics in Computer Science: High Performance Scientific Computing

Students will learn about high performance computing techniques as applied to scientific computations. We will work on Wesleyan's high performance computing (HPC) cluster, gaining familiarity with the HPC environment including the BASH shell and the multiuser queueing system for job submissions. We will design and implement high performance python code, including how to take advantage of parallel processing in central processing units (CPUs) and graphical processing units (GPUs).

We will consider the AMBER molecular dynamics package as a case study of a professional high performance computing package illustrating highly performant code optimized at many levels, and survey a selection of other packages for comparison. The final project will enable students to explore the major themes of the course as applied to their particular areas of interest.

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **None**

Prereq: **COMP212 AND MATH228**

COMP360H Topics in Computer Science: Static Program Analysis

Static program analysis refers to a collection of techniques for analyzing source code to predict execution behavior. It is used for areas such as compiler optimization, programming error detection, and security analysis. This course will focus on the latter two areas, to explore fundamental techniques and concepts. There will be foundational and programming work centered around a medium-scale project.

This course may be used to satisfy the PL Core Area requirement for the COMP major.

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **COMP212 AND MATH228**

COMP361 Advanced Topics in Computer Science

This course covers advanced topics in Computer Science. The precise topics will vary with the offering, but will typically have prerequisites beyond COMP 211-212. This course may be repeated for credit.

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Prereq: **MATH228 AND COMP212**

COMP401 Individual Tutorial, Undergraduate

Topic to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP402 Individual Tutorial, Undergraduate

Topic to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP403 Department/Program Project or Essay

Project to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **A-F**

COMP404 Department/Program Project or Essay

Project to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **A-F**

COMP407 Senior Tutorial (downgraded thesis)

Downgraded Senior Thesis Tutorial - Project to be arranged in consultation with the tutor. Only enrolled in through the Honors Coordinator.

Offering: **Host**

Grading: **A-F**

COMP408 Senior Tutorial (downgraded thesis)

Downgraded Senior Thesis Tutorial - Project to be arranged in consultation with the tutor. Only enrolled in through the Honors Coordinator.

Offering: **Host**

Grading: **A-F**

COMP409 Senior Thesis Tutorial

Topic to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP410 Senior Thesis Tutorial

Topic to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP411 Group Tutorial, Undergraduate

Topic to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP412 Group Tutorial, Undergraduate

Topic to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP419 Student Forum

Student-run group tutorial, sponsored by a faculty member and approved by the chair of a department or program.

Offering: **Host**

Grading: **Cr/U**

COMP420 Student Forum

Student-run group tutorial, sponsored by a faculty member and approved by the chair of a department or program.

Offering: **Host**

Grading: **Cr/U**

COMP421 Undergraduate Research, Sciences

Individual research projects for undergraduate students supervised by faculty members.

Offering: **Host**

Grading: **A-F**

COMP422 Undergraduate Research, Sciences

Individual research projects for undergraduate students supervised by faculty members.

Offering: **Host**

Grading: **OPT**

COMP423 Advanced Research Seminar, Undergraduate

Advanced research tutorial; project to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP424 Advanced Research Seminar, Undergraduate

Advanced research tutorial; project to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP465 Education in the Field, Undergraduate

Students must consult with the department and class dean in advance of undertaking education in the field for approval of the nature of the responsibilities and method of evaluation.

Offering: **Host**

Grading: **OPT**

COMP468 Independent Study, Undergradua

Offering: **Host**

Grading: **OPT**

COMP469 Education in the Field, Undergraduate

Students must consult with the department and class dean in advance of undertaking education in the field for approval of the nature of the responsibilities and method of evaluation.

Offering: **Host**

Grading: A-F

Credits: 1.00

Gen Ed Area: None

Prereq: None

COMP491 Teaching Apprentice Tutorial

The teaching apprentice program offers undergraduate students the opportunity to assist in teaching a faculty member's course for academic credit.

Offering: Host

Grading: OPT

COMP492 Teaching Apprentice Tutorial

The teaching apprentice program offers undergraduate students the opportunity to assist in teaching a faculty member's course for academic credit.

Offering: Host

Grading: OPT

COMP500 Automata Theory and Formal Languages

This course is an introduction to formalisms studied in computer science and mathematical models of computing machines. The language formalisms discussed will include regular, context-free, recursive, and recursively enumerable languages. The machine models discussed include finite-state automata, pushdown automata, and Turing machines.

Offering: Crosslisting

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Identical With: COMP301

Prereq: COMP211 AND COMP212 AND MATH228

COMP501 Individual Tutorial, Graduate

Topic to be arranged in consultation with the tutor.

Offering: Host

Grading: OPT

COMP502 Individual Tutorial, Graduate

Topic to be arranged in consultation with the tutor.

Offering: Host

Grading: OPT

COMP503 Selected Topics, Graduate Sciences

Topic to be arranged in consultation with the tutor. A seminar primarily concerned with papers taken from current research publications designed for, and required of, graduate students.

Offering: Host

Grading: A-F

COMP504 Selected Topics, Graduate Sciences

Topic to be arranged in consultation with the tutor. A seminar primarily concerned with papers taken from current research publications designed for, and required of, graduate students.

Offering: Host

Grading: A-F

COMP510 Algorithms and Complexity

The course will cover the design and analysis of efficient algorithms. Basic topics will include greedy algorithms, divide-and-conquer algorithms, dynamic programming, and graph algorithms. Some advanced topics in algorithms may be selected from other areas of computer science.

Offering: Crosslisting

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Identical With: COMP312

Prereq: COMP212 AND MATH228

COMP511 Group Tutorial, Graduate

Topic to be arranged in consultation with the tutor.

Offering: Host

Grading: OPT

COMP512 Group Tutorial, Graduate

Topic to be arranged in consultation with the tutor.

Offering: Host

Grading: OPT

COMP521 Design of Programming Languages

This course is an introduction to concepts in programming languages. Topics include parameter passing, type checking and inference, control mechanisms, data abstraction, module systems, and concurrency. Basic ideas in functional, object-oriented, and logic programming languages will be discussed.

Offering: Crosslisting

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Identical With: COMP321

Prereq: COMP212 AND MATH228

COMP523 Programming Language Implementation

This course is an introduction to the implementation of programming languages.

Students will learn how to formally describe and implement major components of the implementation pipeline. Topics may include lexical analysis and parsing (checking whether source code is well-formed and converting it to an internal programmatic representation), type-checking and -inference (static program analysis for safety features), interpretation (direct execution of a high-level language program), and compilation (translation to a low-level language such as assembly or bytecode).

Offering: Crosslisting

Grading: OPT

Credits: 1.00

Gen Ed Area: NSM-MATH

Identical With: COMP323

Prereq: COMP212 AND MATH228

COMP527 Evolutionary and Ecological Bioinformatics

Bioinformatic analysis of gene sequences and gene expression patterns has added enormously to our understanding of ecology and evolution. For example, through bioinformatic analysis of gene sequences, we can now reconstruct the evolutionary history of physiology, even though no traces of physiology exist in the fossil record. We can determine the adaptive history of one gene and all the gene's descendants. We can now construct the evolutionary tree of all of life. Bioinformatics is particularly promising for analysis of the ecology and biodiversity of microbial communities, since well over 99 percent of microorganisms cannot be cultured; our only knowledge of these organisms is through analysis of their gene sequences and gene expression patterns. For example, even when we cannot culture most of a microbial community, we can determine which metabolic pathways are of greatest significance through analysis of community-level gene expression. All these research programs are made accessible not only by breakthroughs in molecular technology but also by innovation in the design of computer algorithms. This course, team-taught by an evolutionary biologist and a computer scientist, will present how bioinformatics is revolutionizing evolutionary and ecological investigation and will present the design and construction of bioinformatic computer algorithms underlying the revolution in biology. Students will learn algorithms for reconstructing phylogeny, for sequence alignment, and for analysis of genomes, and students will have an opportunity to create their own algorithms.

Offering: Crosslisting

Grading: A-F

Credits: 1.00

Gen Ed Area: **NSM-BIOL**

Identical With: **BIOL327, COMP327, BIOL527, CIS327**

Prereq: **[BIOL182 or MB&B182] OR [BIOL196 or MBB196] OR COMP112 OR COMP211**

COMP531 Computer Structure and Organization

The purpose of the course is to introduce and discuss the structure and operation of digital computers. Topics will include the logic of circuits, microarchitectures, microprogramming, conventional machine architectures, and an introduction to software/hardware interface issues. Assembly language programming will be used to demonstrate some of the basic concepts.

Offering: **Crosslisting**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **NSM-MATH**

Identical With: **COMP331**

Prereq: **COMP212**

COMP549 Advanced Research Seminar, Graduate

Advanced research tutorial; project to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP550 Advanced Research Seminar, Graduate

Advanced research tutorial; project to be arranged in consultation with the tutor.

Offering: **Host**

Grading: **OPT**

COMP571 Special Topics in Computer Science

Supervised reading course of varying length. This course may be repeated for credit.

Offering: **Host**

Grading: **A-F**

Credits: **1.00**

Gen Ed Area: **None**

Prereq: **None**

COMP572 Special Topics in Computer Science

Supervised reading course of varying length. This course may be repeated for credit.

Offering: **Host**

Grading: **OPT**

Credits: **1.00**

Gen Ed Area: **None**

Prereq: **None**