COMPUTER SCIENCE (CMPS)

CMPS 1005 Python Programming (3)
An introductory course on computer programming, in which the students design, implement, test, and debug programs for computational problems using Python programming language. This course emphasizes program design process, object-oriented software development approach, and practical programming skills that translate to programming in other modern languages. Assignments include practical problems drawn from various fields (e.g. biology, linguistics, graphics, and games). Open to high-school students only, no prerequisites. Credits don't count toward Coordinate Major in Computer Science.

CMPS 1100 Foundations of Programming (3)
An introductory practice-oriented course on computer programming, in which the students design, implement, test and debug programs for computational problems drawn from various fields using Python programming language, while working individually and in groups. This course emphasizes program design process, object-oriented software development approach, and development of practical programming skills that translate to programming in other modern languages. This is a stand-alone introductory computer science course that is not a part of the coordinate major in computer science program. It is aimed at students with no prior computing background who wish to learn the foundations of programming and computational problem solving.

CMPS 1500 Intro to Computer Science I (4)
Computational tools are a critical part of our everyday lives. Software is the driving force behind cutting-edge scientific discovery, blockbuster entertainment, and today's fast-paced marketplace. This course is an introduction to techniques, ideas, and problem-solving approaches that are used to develop some of these tools. At a high level, we focus on developing "computational thinking", which is the practice of using abstraction to design and implement algorithms and software to solve problems that arise in many different areas of our daily lives, such as networks, social media, and scientific computing, to name just a few. At a practical level, students will design, implement, test and document their programs to learn introductory programming concepts, such as: data types and data structures (e.g. lists, dictionaries, trees); programming techniques (modular design using functions, recursion, object-oriented programming); performance analysis via theoretical estimate, profiling and timing. Most assignments in this course are programming assignments aimed to teach the students to express their ideas in efficient and elegant code; no prior programming experience is necessary to join and succeed in the course. Lecture periods are dedicated to introducing new material, discussions, individual and group activities. Lab periods are used for programming practice. CMPS 1500 is the first course for the Coordinate Major in Computer Science. Corequisite(s): CMPS 1501.

Corequisite(s): CMPS 1501.

CMPS 1501 Intro to Computer Sci I Lab (0)
Corequisite lab of CMPS 1500.

Corequisite(s): CMPS 1500.

CMPS 1600 Intro to Computer Science II (4)
This is the second course in the introductory course sequence for Coordinate Major in Computer Science and is the continuation of CMPS 1500. While CMPS 1500 focuses on a broad array of topics in computer science and a single language (Python), this class focuses on several core topics in the design, analysis and implementation of computational tools that are drawn from the fields of data structures, software engineering, and programming languages (such as Java, C, Haskell): object-oriented programming, test-driven development; data structures and abstract data types; imperative programming and memory management; functional programming. By solving practical, real-life problems in different programming languages and in different ways, students learn to select a language and approach most appropriate for the situation, and prepare to learn new languages independently. The high-level goal of this course is to train students to be able to draw from a versatile set of skills, which in turn will provide a strong foundation for further study in computer science. Prerequisite(s): CMPS 1500. Corequisite(s): CMPS 1601.

Prerequisite(s): CMPS 1500.
Corequisite(s): CMPS 1601.

CMPS 1601 Intro to Comp Science II Lab (0)
Corequisite lab of CMPS 1600. Prerequisite(s): CMPS 1500.

Prerequisite(s): CMPS 1500.
Corequisite(s): CMPS 1600.

CMPS 1660 Special Topics in Computer Sci (1-3)
Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 1940 Intro Topics in Computer Sci (1-4)
Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 1950 Computer Science Principles (3)

CMPS 2120 Programming and Prob Solving (3)
An introductory course on computer programming and problem-solving using computers. In this course students design, implement, test and debug programs for computational problems using Python programming language. This course emphasizes program design process, object-oriented software development approach, and practical programming skills that translate to programming in other modern languages. Assignments include practical problems drawn from various fields (such as biology, linguistics, graphics, and games).

CMPS 2170 Intro to Discrete Math (3)
This course is an introduction to several areas of mathematics that are particularly useful in computer science. The topics include an introduction to predicate and propositional logic, mathematical induction, combinatorics and counting, and discrete probability theory. Prerequisite(s): MATH 1210 or MATH 1310 or MATH 1150 or MATH 1110. In lieu of prerequisites please contact instructor for consideration.

Prerequisite(s): MATH 1210, 1310, 1150 or 1110.

CMPS 2200 Intro to Algorithms (3)
This course is an introduction to the design and analysis of algorithms, and covers several basic algorithmic paradigms and their application to core computational problems in graph theory and optimization, as well as analysis of time and space complexity. The primary focus of the course will be on understanding the divide-and-conquer, greedy and dynamic programming paradigms for algorithm design as well as the problem areas to which they can be applied. Example application areas include graph theory, discrete optimization, numeric and scientific computing and machine learning. Topics usually include: asymptotic analysis and big-O notation; divide-and-conquer algorithms; recurrences and the master method; greedy algorithms; graph algorithms (Breadth-First Search, Depth-First Search, Connectivity and Shortest Paths); dynamic programming; linear programming; lower bounds and computational complexity. Prerequisite(s): CMPS 1600 and (MATH 2170 or CMPS 2170). Corequisite(s): CMPS 2201.

Prerequisite(s): CMPS 1600 and (MATH 2170 or CMPS 2170).
Corequisite(s): CMPS 2201.

CMPS 2201 Intro to Algorithms Lab (0)
Corequisite lab of CMPS 2200. Prerequisite(s): CMPS 1600 and (MATH 2170 or CMPS 2170). Corequisite(s): CMPS 2200.

Prerequisite(s): CMPS 1600 and (MATH 2170 or CMPS 2170).
Corequisite(s): CMPS 2200.

CMPS 2300 Intro to Comp Sys & Networking (3)
Modern computer systems must take advantage not only of the latest hardware technology, but also of the ability to compute and communicate over a network. In this course the students will study the principles behind the design of modern operating systems and distributed systems through theoretical study of classic solutions and hands-on programming assignments in C. The study of architecture and organization of modern operating systems focuses on the concepts of virtualization, concurrency, and persistence. In the study of distributed systems we will examine topics such as protocol design, asynchronous and synchronous communication, and layered network architecture. Prerequisite(s): CMPS 1600. Corequisite(s): CMPS 2301.

Prerequisite(s): CMPS 1600.
Corequisite(s): CMPS 2301.

CMPS 2301 Intro to Comp Sys & Netwk Lab (0)
Corequisite lab of CMPS 2300. Prerequisite(s): CMPS 1600.

Prerequisite(s): CMPS 1600.
Corequisite(s): CMPS 2300.

CMPS 2940 Transfer Coursework (0-20)
Transfer Coursework at the 2000 level. Department approval may be required.

Maximum Hours: 99

CMPS 3130 Intro Comp Geom (3)
This course provides an introduction to geometric algorithms and geometric data structures. Computational Geometry is a young discipline which enjoys close relations to mathematics and to various application areas such as geometric databases, molecular biology, sensor networks, visualization, geographic information systems (GIS), VLSI, robotics, computer graphics and geometric modeling. Covered topics include fundamental geometric algorithm design and analysis paradigms, geometric data structures for planar subdivisions and range searching, algorithms to compute the convex hull, Voronoi diagrams, and Delaunay triangulation, as well as selected advanced topics. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.

CMPS 3140 Intro Artificial Intelligence (3)
The aim of this course is to provide the student with an introduction to the main concepts and techniques playing a key role in the modern arena of artificial intelligence. In addition to covering the main topics that concern modern AI, particular attention will be devoted to its applications in several fields. Among the topics covered are: "What is an intelligent artificial agent?", problem solving using search and constraint satisfaction, uncertainty, Bayesian networks and probabilistic inference, supervised learning, planning, sequential decision problems, as well as several additional topics. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.

CMPS 3160 Introduction to Data Science (3)
The aim of this course is to provide the student with an introduction to the main concepts and techniques required for collecting, processing, and deriving insight into data. Data Science is an interdisciplinary set of topics that includes everything you need to create data driven answers and solutions to specific business, scientific, or sociological questions. Topics typically covered include an introduction to one or more data collection and management systems, e.g., SOL, web scraping, and various data repositories; exploratory and statistical data analysis, e.g., bootstrapping, measures of central tendency, hypothesis testing and machine learning techniques including linear regression and clustering; data and information visualization, e.g., plotting and interactive charts using various technologies; and presentation and communication of the results of these analyses. Prerequisite(s): CMPS 1600 and (CMPS 2170 or MATH 2170).

Prerequisite(s): CMPS 1600 and (CMPS 2170 or MATH 2170).

CMPS 3210 Algs Comp Struct Bio (3)
Over the last few decades, as we have been able to determine whole genome sequences, structural biologists have sought to determine and catalog protein structures with an increasing reliance on computational methods. Automated methods to analyze protein structure make it possible to leverage information from previously solved structures, and to interpret experimental data in a principled way. In this course, we will focus on the myriad of algorithms for analyzing numerous aspects of protein structure and protein-protein interactions. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.

CMPS 3240 Intro to Machine Learning (3)
This course provides an introduction to the fundamental concepts of machine learning and statistical pattern recognition. In addition, several examples of applications will be described. The topics covered include generative/discriminative and parametric/non-parametric supervised learning, including neural networks; unsupervised learning, including clustering, dimensionality reduction and kernel methods; learning theory, including tradeoffs, large margins and VC theory; reinforcement learning, including criteria for optimality, brute force methods, value function methods and direct policy search; feedforward/feedback adaptive control, direct/indirect adaptive control methods; and various applications. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.

CMPS 3250 Theory of Computation (3)
This course is an introduction to the theory of computation. It begins with regular languages and their representation as finite state automata, and continues with context free languages and pushdown automata. Turing machines and the Church-Turing Thesis are also considered, as well as decidability and reducibility. The basic notions of complexity theory area also covered, including P and NP for time complexity, as well as basic results about space complexity. Prerequisite(s): MATH 2170 or CMPS 2170.

Prerequisite(s): MATH 2170 or CMPS 2170.

CMPS 3260 Advanced Algorithms (3)
This course focuses on advanced techniques in the design and analysis of algorithms and illustrates how they are used in deriving a variety of now-classic results. Topics include graph algorithms, randomized algorithms, parallel computing, linear programming, and approximation algorithms. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.

CMPS 3280 Information Theory (3)
This course is an introduction to Shannon's mathematical theory of information. It considers basic concepts such as information content, entropy and the Kullback-Leibler distance, as well as areas such as data compression and Shannon's Source Coding Theorem, coding, prefix codes, lossless channels and their capacity, and Shannon's Noisy Coding Theorem. Applications to various areas are also featured in the course. Prerequisite(s): MATH 3050 or 3090.

Prerequisite(s): MATH 3050 or 3090.

CMPS 3300 Software Studio (3)
This is a project-oriented course on fundamentals of software development and software engineering. Working in teams, students apply a recognized software engineering methodology, a modern programming language, and software development tools (including an IDE, debugger, version control system, and testing framework) to design and implement a semester-long project – a software solution for a real-world problem. The high goal of the course is to train students to function efficiently in a real-world software development environment. To help reach that goal, the students do a lot of independent learning, teamwork, documentation and public presentation of their product and design process. The particular technologies employed in the course may change in synchrony with changes in the software engineering field, currently the focus is on engineering software-as-a-service using Ruby for programming language and Rails for web development framework. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.

CMPS 3310 Logic in Computer Science (3)
This course is an introduction to logic and its applications in computer science. The topics covered include soundness and completeness of propositional logic, predicate logic, linear time temporal logic and branching time temporal logics, and their expressive power, frameworks for software verification, Hoare triples, partial and total correctness, modal logics and agents, and binary decision diagrams. Prerequisite(s): CMPS 2200 and (CMPS 2170 or MATH 2170).

Prerequisite(s): CMPS 2200 and (CMPS 2170 or MATH 2170).

CMPS 3350 Intro to Computer Graphics (3)
A comprehensive introduction to the mathematics and algorithms that drive today's digital special effects, animation, and games. Designed as a hands-on course, students will gain experience in building 2D/3D interactive applications using OpenGL. Topics covered will include geometric transformations, projections, raster algorithms, 3D object models (surface and volume), visible surface algorithms, texture mapping, lighting/shading, ray-tracing, anti-aliasing, and compositing. Prerequisite(s): CMPS 1600.

Prerequisite(s): CMPS 1600.

CMPS 3360 Data Visualization (3)
An introduction on how graphical representations of data can be used to aid understanding. This course details the theory and practice of designing effective information or scientific visualizations. The techniques learned in this class have wide applications to all fields in engineering and science, where due to increasing sizes and complexity, data now demands effective presentation and analysis. Topics will include iso-surfacing, volume rendering, transfer functions, vector/tensor fields, topological analysis, large data visualization, and uncertainty in visualizations. Prerequisite(s): CMPS 1600.

Prerequisite(s): CMPS 1600.

CMPS 3660 Special Topics in Computer Sci (1-3)
This course varies from time to time, focusing on topics of interest to the faculty and students. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 3665 Special Topics Lab (1-3)
Special Topics Lab. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 3890 Service Learning (0-1)
Students complete a service activity in the community in conjunction with the content of a three-credit co-requisite course. Course may be repeated up to unlimited credit hours.

Corequisite(s): CMPS 3160.
Maximum Hours: 99

CMPS 3940 Transfer Coursework (0-20)
Transfer Coursework at the 3000 level. Department approval may be required.

Maximum Hours: 99

CMPS 4010 Capstone Project I (2)
This is the first semester of a two-semester course devoted to the development of the student's capstone project, a required component of the Computer Science coordinate major. Under supervision of a faculty advisor in computer science, students use the tools of computer science to solve a problem from another discipline, usually their primary major area. Prerequisite(s): CMPS 2200 and 2300.

Prerequisite(s): CMPS 2200 and 2300.

CMPS 4020 Capstone Project II (2)
This is the second of a two-semester course devoted to the development of the student's capstone project, a required component of the Computer Science coordinate major. Under supervision of a faculty advisor in computer science, students use the tools of computer science to solve a problem from another discipline, usually their primary major area. Prerequisite(s): CMPS 4010.

Prerequisite(s): CMPS 4010.

CMPS 4150 Multi-agent Systems (3)
This course has two main goals. The first one is to give a broad overview of the fundamentals of multi-agent systems (MAS). MAS are playing an increasingly important role in Artificial Intelligence as distributed resources push for highly distributed forms of intelligence. The second aim is to provide a more in-depth discussion of selected MAS topics: game theory and voting from a computational point of view. Situated at the nexus between economics and computer science, these research areas provide a perfect example of interdisciplinary cross-fertilization and mutual enrichment and lie at the core of multi-agent systems theory. The course will provide the student with an understanding of how self-interested behavior and coordination can be formally modeled and implemented in societies of artificial agents. Course may be repeated up to unlimited credit hours. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.
Maximum Hours: 99

CMPS 4250 Math Found Comp Security (3)
This course studies the mathematics underlying computer security, including both public key and symmetric key cryptography, crypto-protocols and information flow. The course includes a study of the RSA encryption scheme, stream and clock ciphers, digital signatures and authentication. It also considers semantic security and analysis of secure information flow. Prerequisite(s): (MATH 1160 or 1220) and (MATH 2170 or CMPS 2170).

Prerequisite(s): (MATH 1220 or 1310) and (MATH 2170 or CMPS 2170).

CMPS 4610 Algorithms (3)
This course covers fundamental algorithm design principles and data structures, basic notions of complexity theory, as well as an advanced introduction to parallel algorithms, randomized algorithms, and approximation algorithms. Topics include: divide-and-conquer, dynamic programming, amortized analysis, graph algorithms, network flow, map reduce, and more advanced topics in approximation algorithms and randomized algorithms. Prerequisite(s): (CMPS 2170 or MATH 2170) and (CMPS 2200).

Prerequisite(s): (CMPS 2170 or MATH 2170) and (CMPS 2200).

CMPS 4620 Artificial Intelligence (3)
This course is designed for graduate students interested in understanding the design of autonomous intelligent agents. The course will cover fundamental notions and concepts such as uninformed and informed search, local search, constraint satisfaction and constraint-based optimization, Bayesian Networks, Markov Decision Problems and a short introduction on machine learning. Furthermore, advance topics and applications in the context of natural language processing, reasoning about time, algorithmic game theory and computational social choice will be considered as well. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.

CMPS 4630 Computational Bio & Bioinform (3)
This course is an introduction to computational methods in molecular biology. Topics covered include: sequence analysis and alignment, sequencing technologies, genome and metagenomic sequencing, protein structure and structure prediction, and phylogenetic analysis. No prior background in biology is assumed. Prerequisite(s): (CMPS 2170 or MATH 2170) and CMPS 2200.

Prerequisite(s): (CMPS 2170 or MATH 2170) and CMPS 2200.

CMPS 4640 Adv. Computational Geometry (3)
This course focuses on advanced principles for designing and analyzing geometric algorithms and data structures, and their application to other disciplines. Selected topics may include: Dynamic and kinetic data structures, geometric algorithms and data structures in two and higher dimensions, shape analysis and matching, robustness and implementation issues, geometric approximation algorithms. Prerequisites: Introduction to Algorithms or equivalent, or permission by the instructor. CMPS 3130/6130 preferred. Prerequisite(s): CMPS 2200.

Prerequisite(s): CMPS 2200.

CMPS 4660 Special Topics (1-3)
Special topics in Computer Science. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 4710 Computational Complexity (3)
This course is an advanced introduction to the area of computational complexity. Topics covered include: impossibility and separability results for classical computation, interactive theorem proving and the PCP theorem, derandomization and hardness of approximation, and the quantum model of computation. Prerequisite(s): CMPS 3260, MATH 3260, CMPS 4610, 6610 or 3250.

Prerequisite(s): CMPS 3260, MATH 3260, CMPS 4610, 6610 or 3250.

CMPS 4720 Machine Learning (3)
This course will cover fundamental and advanced topics in machine learning. Topics will include linear and logistic regression, Lasso, preceptrons, deep neural networks, support vector machines, kernel methods, graphical models, principal and independent component analysis and Gaussian processes. In addition to thoroughly addressing theoretical aspects, several examples will illustrate the application of the different techniques. Prerequisite(s): (CMPS 2170 or MATH 2170) and CMPS 2200.

Prerequisite(s): (CMPS 2170 or MATH 2170) and CMPS 2200.

CMPS 4730 Natural Language Processing (3)
This course investigates computational methods to work with human language, analyzing its lexical, syntactic, and semantic aspects. Examples include document classification and clustering, syntactic parsing, information extraction, speech recognition, and machine translation. Theoretical and practical aspects of the latest techniques will be covered, including probabilistic modeling, neural networks, and deep learning. Prerequisite(s): CMPS 3140 or CMPS 3160 or CMPS 3240.

Prerequisite(s): CMPS 3140, 3160 or 3240.

CMPS 4750 Computer Networks (3)
The objective of the course is to introduce students to the core concepts and analytic techniques in the design and analysis of computer networks and network protocols. We will explain both how computer networks work using the Internet as the paradigm and why they work from an optimization and control perspective. Prerequisite(s): (CMPS 2170 or MATH 2170) and CMPS 2200.

Prerequisite(s): (CMPS 2170 or MATH 2170) and CMPS 2200.

CMPS 4760 Distributed Systems (3)
This course covers the fundamental concepts in distributed computing. The objective is to introduce students to the core notions, algorithms, and analytic tools in the design of distributed systems. Recent developments in peer-to-peer systems, cloud computing, sensor networks, etc. will be used as case studies to help students establish a firm understanding of the philosophy and pitfalls in the design of computer systems when there is no global clock and when unpredictable failures and variable latency are the norm. Prerequisite(s): (MATH 2170 or CMPS 2170) and CMPS 2200 and 2300.

Prerequisite(s): (MATH 2170 or CMPS 2170) and CMPS 2200 and 2300.

CMPS 4910 Independent Study (1-3)
This is a directed study course that allows a student to pursue a topic of particular interest under the direction of a computer science faculty member. No more than three hours of 4910-4920 may be counted toward satisfying the major requirements. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 4920 Independent Study (1-3)
This is a directed study course that allows a student to pursue a topic of particular interest under the direction of a computer science faculty member. No more than three hours of 4910-4920 may be counted toward satisfying the major requirements.

CMPS 4940 Transfer Coursework (0-20)
Transfer coursework at the 4000 level. Departmental approval required.

Maximum Hours: 99

CMPS 4990 Honors Thesis (3)

CMPS 5000 Honors Thesis (4)

CMPS 5380 Study Abroad (1-20)
Courses taught abroad by non-Tulane faculty. Does not count toward Tulane GPA. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 5390 Study Abroad (1-20)
Courses taught abroad by non-Tulane faculty. Does not count toward Tulane GPA. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 6130 Intro Comp Geom (3)
This course provides an introduction to geometric algorithms and geometric data structures. Computational Geometry is a young discipline which enjoys close relations to mathematics and to various application areas such as geometric databases, molecular biology, sensor networks, visualization, geographic information systems (GIS), VLSI, robotics, computer graphics and geometric modeling. Covered topics include fundamental geometric algorithm design and analysis paradigms, geometric data structures for planar subdivisions and range searching, algorithms to computer the convex hull, Voronoi diagrams, and Delaunay triangulation, as well as selected advanced topics.

CMPS 6140 Intro Artificial Intelligence (3)
The aim of this course is to provide the student with an introduction to the main concepts and techniques playing a key role in the modern arena of artificial intelligence. In addition to covering the main topics that concern modern AI, particular attention will be devoted to its applications in several fields. Among the topics covered are: "What is an intelligent artificial agent?", problem solving using search and constraint satisfaction, uncertainty, Bayesian networks and probabilistic inference, supervised learning, planning, sequential decision problems, as well as several additional topics.

CMPS 6150 Multi-agent Systems (3)
This course has two main goals. The first one is to give a broad overview of the fundamentals of multi-agent systems (MAS). MAS are playing an increasingly important role in Artificial Intelligence as distributed resources push for highly distributed forms of intelligence. The second aim is to provide a more in depth discussion of selected MAS topics: game theory and voting from a computational point of view. Situated at the nexus between economics and computer science, these research areas provide a perfect example of interdisciplinary cross-fertilization and mutual enrichment and lie at the core of multi-agent systems theory. The course will provide the student with an understanding of how self-interested behavior and coordination can be formally modeled and implemented in societies of artificial agents. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 6160 Introduction to Data Science (3)
The aim of this course is to provide the student with an introduction to the main concepts and techniques required for collecting, processing, and deriving insight into data. Data Science is an interdisciplinary set of topics that includes everything you need to create data driven answers and solutions to specific business, scientific, or sociological questions. Topics typically covered include an introduction to one or more data collection and management systems, e.g., SOL, web scraping, and various data repositories; exploratory and statistical data analysis, e.g., bootstrapping, measures of central tendency, hypothesis testing and machine learning techniques including linear regression and clustering; data and information visualization, e.g., plotting and interactive charts using various technologies; and presentation and communication of the results of these analyses.

CMPS 6210 Algs Comp Struct Bio (3)
Over the last few decades, as we have been able to determine whole genome sequences, structural biologists have sought to determine and catalog protein structures with an increasing reliance on computational methods. Automated methods to analyze protein structure make it possible to leverage information from previously solved structures, and to interpret experimental data in a principled way. In this course, we will focus on the myriad of algorithms for analyzing numerous aspects of protein structure and protein-protein interactions.

CMPS 6240 Intro to Machine Learning (3)
This course provides an introduction to the fundamental concepts of machine learning and statistical pattern recognition. In addition, several examples of applications will be described. The topics covered include generative/discriminative and parametric/non-parametric supervised learning, including neural networks; unsupervised learning, including clustering, dimensionality reduction and kernel methods; learning theory, including tradeoffs, large margins and VC theory; reinforcement learning, including criteria for optimality, brute force methods, value function methods and direct policy search; feedforward/feedback adaptive control, direct/indirect adaptive control methods; and various applications.

CMPS 6250 Math Found Comp Security (3)
This course studies the mathematics underlying computer security, including both public key and symmetric key cryptography, crypto-protocols and information flow. The course includes a study of the RSA encryption scheme, stream and clock ciphers, digital signatures and authentication. It also considers semantic security and analysis of secure information flow.

CMPS 6260 Advanced Algorithms (3)
This course focuses on advanced techniques in the design and analysis of algorithms and illustrates how they are used in deriving a variety of now-classic results. Topics include graph algorithms, randomized algorithms, parallel computing, linear programming and approximation algorithms.

CMPS 6280 Information Theory (3)
This course is an introduction to Shannon's mathematical theory of information. It considers basic concepts such as information content, entropy and the Kullback-Leibler distance, as well as areas such as data compression and Shannon's Source Coding Theorem, coding, prefix codes, lossless channels and their capacity, and Shannon's Noisy Coding Theorem. Applications to various areas are also featured in the course.

CMPS 6300 Software Studio (3)
This is a project-oriented course on fundamentals of software development and software engineering. Working in teams, students apply a recognized software engineering methodology, a modern programming language, and software development tools (including an IDE, debugger, version control system, and testing framework) to design and implement a semester-long project – a software solution for a real-world problem. The high goal of the course is to train students to function efficiently in a real-world software development environment. To help reach that goal, the students do a lot of independent learning, teamwork, documentation and public presentation of their product and design process. The particular technologies employed in the course may change in synchrony with changes in the software engineering field, currently the focus is on engineering software-as-a-service using Ruby for programming language and Rails for web development framework.

CMPS 6310 Logic in Computer Science (3)
This course is an introduction to logic and its applications in computer science. The topics covered include soundness and completeness of propositional logic, predicate logic, linear time temporal logic and branching time temporal logics and their expressive power, frameworks for software verification, Hoare triples, partial and total correctness, modal logics and agents, and binary decision diagrams.

CMPS 6350 Intro to Computer Graphics (3)
A comprehensive introduction to the mathematics and algorithms that drive today's digital special effects, animation, and games. Designed as a hands-on course, students will gain experience in building 2D/3D interactive applications using OpenGL. Topics covered will include geometric transformations, projections, raster algorithms, 3D object models (surface and volume), visible surface algorithms, texture mapping, lighting/shading, ray-tracing, anti-aliasing, and compositing.

CMPS 6360 Data Visualization (3)
An introduction on how graphical representations of data can be used to aid understanding. This course details the theory and practice of designing effective information or scientific visualizations. The techniques learned in this class have wide applications to all fields in engineering and science, where due to increasing sizes and complexity, data now demands effective presentation and analysis. Topics will include iso-surfacing, volume rendering, transfer functions, vector/tensor fields, topological analysis, large data visualization, and uncertainty in visualizations.

CMPS 6610 Algorithms (3)
This course covers fundamental algorithm design principles and data structures, basic notions of complexity theory, as well as an advanced introduction to parallel algorithms, randomized algorithms, and approximation algorithms. Topics include: divide-and-conquer, dynamic programming, amortized analysis, graph algorithms, network flow, map reduce, and more advanced topics in approximation algorithms and randomized algorithms.

CMPS 6620 Artificial Intelligence (3)
This course is designed for graduate students interested in understanding the design of autonomous intelligent agents. The course will cover fundamental notions and concepts such as uninformed and informed search, local search, constraint satisfaction and constraint-based optimization, Bayesian Networks, Markov Decision Problems and a short introduction on machine learning. Furthermore, advance topics and applications in the context of natural language processing, reasoning about time, algorithmic game theory and computational social choice will be considered as well.

CMPS 6630 Computational Bio & Bioinform (3)
This course is an introduction to computational methods in molecular biology. Topics covered include: sequence analysis and alignment, sequencing technologies, genome and metagenomic sequencing, protein structure and structure prediction, and phylogenetic analysis. No prior background in biology is assumed.

CMPS 6640 Adv. Computational Geometry (3)
This course focuses on advanced principles for designing and analyzing geometric algorithms and data structures, and their application to other disciplines. Selected topics may include: Dynamic and kinetic data structures, geometric algorithms and data structures in two and higher dimensions, shape analysis and matching, robustness and implementation issues, geometric approximation algorithms. Prerequisites: Introduction to Algorithms or equivalent, or permission by the instructor. CMPS 3130/6130 preferred.

Prerequisite(s): CMPS 2200.

CMPS 6660 Special Topics in Computer Sci (1-3)
This course varies from time to time, focusing on topics of interest to the faculty and students. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 6710 Computational Complexity (3)
This course is an advanced introduction to the area of computational complexity. Topics covered include: impossibility and separability results for classical computation, interactive theorem proving and the PCP theorem, derandomization and hardness of approximation, and the quantum model of computation.

CMPS 6720 Machine Learning (3)
This course will cover fundamental and advanced topics in machine learning. Topics will include linear and logistic regression, Lasso, preceptrons, deep neural networks, support vector machines, kernel methods, graphical models, principal and independent component analysis and Gaussian processes. In addition to thoroughly addressing theoretical aspects, several examples will illustrate the application of the different techniques.

CMPS 6730 Natural Language Processing (3)
This course investigates computational methods to work with human language, analyzing its lexical, syntactic, and semantic aspects. Examples include document classification and clustering, syntactic parsing, information extraction, speech recognition, and machine translation. Theoretical and practical aspects of the latest techniques will be covered, including probabilistic modeling, neural networks, and deep learning.

CMPS 6750 Computer Networks (3)
The objective of the course is to introduce students to the core concepts and analytic techniques in the design and analysis of computer networks and network protocols. We will explain both how computer networks work using the Internet as the paradigm and why they work from an optimization and control perspective.

CMPS 6760 Distributed Systems (3)
This course covers the fundamental concepts in distributed computing. The objective is to introduce students to the core notions, algorithms, and analytic tools in the design of distributed systems. Recent developments in peer-to-peer systems, cloud computing, sensor networks, etc. will be used as case studies to help students establish a firm understanding of the philosophy and pitfalls in the design of computer systems when there is no global clock and when unpredictable failures and variable latency are the norm.

CMPS 6910 Independent Study (3)
This is a directed study course that allows a student to pursue a topic of particular interest under the direction of a computer science faculty member.

CMPS 6940 Transfer Coursework (0-20)
Transfer coursework at the 6000 level. Departmental approval required.

Maximum Hours: 99

CMPS 7010 Research Seminar (3)
This seminar course introduces students to research methods in Computer Science and to the research conducted in the department. Students will read research papers, participate in active research projects, and practice preparing and presenting research presentations. Department faculty will present on their research in order to expose students to the research projects conducted in the department. Research methods such as literature search, experiment design, technical writing, etc. will also be covered. This course is required for all PhD students in Computer Science. The content of this course varies from semester to semester.

CMPS 7020 Research in Computer Science (3)
In this course PhD students engage in a research project in Computer Science, under the direction of a faculty member, normally the student's faculty advisor. At the beginning of the course the scope of the project will be determined by a project proposal prepared by the student. Research will be conducted on an individual basis with the faculty advisor. The student will prepare a final report that summarize the research methodologies and the research outcomes. The content of this course varies from semester to semester.

CMPS 7021 Research in Computer Science (3)
In this course PhD students engage in a research project in Computer Science, under the direction of a faculty member, normally the student's faculty advisor. At the beginning of the course the scope of the project will be determined by a project proposal prepared by the student. Research will be conducted on an individual basis with the faculty advisor. The student will prepare a final report that summarize the research methodologies and the research outcomes. The content of this course varies from semester to semester.

CMPS 7022 Research in Computer Science (3)
In this course PhD students engage in a research project in Computer Science, under the direction of a faculty member, normally the student's faculty advisor. At the beginning of the course the scope of the project will be determined by a project proposal prepared by the student. Research will be conducted on an individual basis with the faculty advisor. The student will prepare a final report that summarize the research methodologies and the research outcomes. The content of this course varies from semester to semester.

CMPS 7120 Advanced Topics in Computer Sc (3)
This course varies from time to time, focusing on advanced topics of interest to the faculty and students.

CMPS 7940 Transfer Coursework (0-20)
Transfer coursework at the 7000 level. Departmental approval required.

Maximum Hours: 99

CMPS 7980 Independent Study (3)
This is a directed study course that allows a graduate student to pursue a topic of particular interest under the direction of a computer science faculty member. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 9980 Masters Research (3)
Research toward completion of a masters degree. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99

CMPS 9990 Dissertation Research (3)
Research toward completion of a doctoral degree. Course may be repeated up to unlimited credit hours.

Maximum Hours: 99