Computer Science Program Course Descriptions

CS 207. Programming Methodology and Abstractions (3-0-3) Computer programming and the use of abstractions. Software engineering principles of data abstraction and modularity. Object-oriented programming, fundamental data structures (such as stacks, queues, sets) and data-directed design. Recursion and recursive data structures (linked lists, trees, graphs). Introduction to basic time and space complexity analysis. The course teaches the mechanics of the C, C++ or Java language. This course is considered remedial training for students in the CS program and will not count toward any degree requirement.

CS 209. Digital Systems (3-0-3) *Prerequisites: facility with at least one programming language (at least at the level of CS 207) and logic.* The design of processor-based digital systems. Instruction sets, addressing modes, data types. Assembly language programming, low-level data structures, introduction to operating systems and compilers. Processor micro architecture, microprogramming, pipelining. Memory systems and caches. Input/output, interrupts, buses and DMA. System design implementation alternatives, software/hardware tradeos. Labs involve the design of processor subsystems and processor-based embedded systems.

CS 210. Applied Probability and Biostatistics (3-0-3) *Prerequisites: Advanced and multivariate calculus.* Probability: random variables, independence, and conditional probability; discrete and continuous distributions, moments, distributions of several random variables. Topics in mathematical statistics: random sampling, point estimation, confidence intervals, hypothesis testing, nonparametric tests, regression and correlation analyses. Applications in engineering, industrial manufacturing, medicine, biology, and other fields.

CS 211. Numerical Optimization (3-0-3) *Recommended prerequisite: AMCS 171*. Solution of nonlinear equations. Optimality conditions for smooth optimization problems. Theory and algorithms to solve unconstrained optimization, linear programming, quadratic programming, global optimization, general linearly and nonlinearly constrained optimization problems. Programming project.

CS 212. Linear and Nonlinear Optimization (3-0-3) Optimization theory and modeling. The role of prices, duality, optimality conditions and algorithms in finding and recognizing solutions. Perspectives: problem formulation, analytical theory, computational methods and recent applications in engineering, finance and economics. Theories: finite dimensional derivatives, convexity, optimality, duality and sensitivity. Methods: simplex and interior-point, gradient, Newton and barrier.

CS 221. Artificial Intelligence (3-0-3) *Prerequisites: working knowledge of basic discrete mathematics (e.g., sets and functions) and proof techniques, programming ability (at least at the level of CS 207) and exposure to probability*. An introduction to the principles and practices of artificial intelligence. Topics include: search, constraint satisfaction, knowledge representation, probabilistic models, machine learning, neural networks, vision, robotics and natural language understanding.

CS 229. Machine Learning (3-0-3) *Prerequisites: linear algebra and basic probability and statistics. Familiarity with artificial intelligence recommended.* Topics: statistical pattern recognition, linear and non-linear regression, nonparametric methods, exponential family, GLIMs, support vector machines, kernel methods, model/ feature selection, learning theory, VC dimension, clustering, density estimation, EM, dimensionality reduction, ICA, PCA, reinforcement learning and adaptive control, Markov decision processes, approximate dynamic programming and policy search.

CS 240. Operating Systems and Systems Programming (3-0-3) *Prerequisite: solid computer programming skills (at least at the level of CS 207).* Operating systems design and implementation. Basic structure; synchronization and communication mechanisms; implementation of processes, process management, scheduling and protection; memory organization and management, including virtual memory; I/O device management, secondary storage and file systems.

CS 241. Probability and Random Process (3-0-3) *Prerequisites: Advanced and multivariate calculus.* Introduction to probability and random processes. Topics include probability axioms, sigma algebras, random vectors, expectation, probability distributions and densities, Poisson and Wiener processes, stationary processes, autocorrelation, spectral density, effects of filtering, linear least squares estimation and convergence of random sequences.

CS 242. Programming Languages (3-0-3) *Prerequisites: programming experience with Lisp, C and an object-oriented programming language.* Central concepts in modern programming languages, impact on software development, language design trade-o s and implementation considerations. Functional, imperative and object oriented paradigms. Formal semantic methods and program analysis. Modern type systems, higher-order functions and closures, exceptions and continuations. Modularity, object-oriented languages, and concurrency. Runtime support for language features, interoperability and security issues.

CS 243. Compilers (3-0-3) *Prerequisites: solid computer programming skills (at least at the level of CS 207), familiarity with formal languages (regular expressions and grammars).* Principles and practices for design and implementation of compilers and interpreters. Topics: lexical analysis, parsing theory, symbol tables, type systems, scope, semantic analysis, intermediate representations, run-time environments, co de generation and basic program analysis and optimization. Students construct a compiler for a simple object-oriented language during course programming projects.

CS 244. Computer Networks (3-0-3) *Prerequisite: Operating systems and systems programming.* Packet switching, Internet architecture, routing, router architecture, control algorithms, retransmission algorithms, congestion control, TCP/IP, detecting and recovering from errors, switching, Ethernet (wired and wireless) and local area networks, physical layers. clocking and synchronization. Assignments intro duce network programming, including sockets, designing a router and implementing a transport layer.

CS 245. Databases (3-0-3) *Prerequisites: working knowledge of basic discrete mathematics (e.g., sets, functions and relations) and programming skills.* Database design and use of database management systems for applications. The relational model, relational algebra and SQL, the standard language for creating, querying and modifying relational and object-relational databases. XML data including the query languages XPath and XQuery. UML database design and relational design principles based on functional dependencies and normal forms. Other topics include indexes, views, transactions, authorization, integrity constraints and triggers. Advanced topics from data warehousing, data mining, Web data management, Datalog, data integration, data streams and continuous queries and data-intensive Web services.

CS 247. Scientific Visualization (3-0-3) *Prerequisites: Advanced and multivariate calculus, and linear algebra, computer graphics, and programming experience.* Techniques for generating images of various types of experimentally measured, computer generated, or gathered data. Grid structures. Scalar field visualization. Vector field visualization. Particle visualization. Graph visualization. Animation. Applications in science, engineering, and medicine.

CS 248. Computer Graphics (3-0-3) *Prerequisites: solid programming skills and linear algebra.* Input and display devices, scan conversion of geometric primitives, 2D and 3D geometric transformations, clipping and windowing, scene modeling and animation, algorithms for visible surface determination, local and global shading models, color and realtime rendering methods.

CS 251. Numerical Linear Algebra (3-0-3) *Prerequisites: Programming skills and linear algebra.* Linear algebra in the presence of floating point rounding error, with applications to large scale scientific computing. Matrix factorizations. Linear least squares. Accuracy and stability. Eigen analysis. Singular Value Decomposition. Krylov subspace methods and preconditioning. Optimization and saddle point systems.

CS 260. Design and Analysis of Algorithms (3-0-3) *Prerequisites: computer programming skills, knowledge of probability, understanding of basic data structures, basic knowledge in discrete mathematics.* Fulfills University Mathematics Requirement. Review of algorithm analysis (search in ordered array, binary insertion sort, merge sort, 2-3 trees, asymptotic notation). Divide and conquer algorithms (master theorem, integer multiplication, matrix multiplication, fast Fourier transform). Graphs (breadth-first search, connected components, topological ordering, depth-first search). Dynamic programming (chain matrix multiplication, shortest paths, edit distance, sequence alignment). Greedy algorithms (binary heaps, Dijkstra's algorithm, minimum spanning tree, Huffman codes). Randomized algorithms (selection, quick sort, global minimum cut, hushing). P and NP (Cook's theorem, examples of NP-complete problems). Approximate algorithms for NP-hard problems (set cover, vertex cover, maximum independent set). Partial recursive functions (theorem of Post, Diophantine equations). Computations and undecidable problems (undecidability of halting problem, theorem of Rice, semantic and syntactical properties of programs).

CS 261. Algorithmic Paradigms (3-0-3) *Prerequisite: Familiarity with discrete algorithms at the level of AMCS 260.* Topics: algorithms for optimization problems such as matching, max flow, min-cut and load balancing. Using linear programming, emphasis is on LP duality for design and analysis of approximation algorithms. Approximation algorithms for NP-complete problems such as Steiner trees, traveling salesman and scheduling problems. Randomized algorithms.

CS 271. Applied Geometry (3-0-3) Differential Geometry: selected topics from the classical theory of curves and surfaces, geometric variational problems, robust computation of differential invariants, discrete differential geometry. Projective Geometry: computing with homogeneous coordinates, projective maps, quadrics and polarity. Algebraic Geometry: algebraic curves and surfaces, rational parametrizations, basic elimination theory. Kinematical Geometry: geometry of motions, kinematic mappings. The practical use of these topics is illustrated at hand of sample problems from Geometric Modeling, Computer Vision, Robotics and related areas of Geometric Computing.

CS 272 Geometric Modeling (3-0-3) *Prerequisites: Advanced and multivariate calculus, and linear algebra, computer graphics, and programming experience.* Terminology, coordinate systems, and implicit forms. Parametric and spline representations of curves and surfaces and their uses. Basic differential geometry of curves and surfaces. Subdivision surfaces. Solid modeling paradigms and operations. Robustness and accuracy in geometric computations. Applications.

CS 282. Computer Architecture and Organization (3-0-3) *Prerequisite: CS 209.* Advanced topics in cache hierarchies, memory systems, storage and IO systems, interconnection networks and message passing multi-processor systems (clusters). Issues such as locality, coarse grain parallelism, synchronization, overlapping communication with computation, hardware/software interfaces, performance/power trade-o s and reliability. Characteristics of modern processors that affect system architecture.

CS 291 Scientific Software Engineering (3-0-3) *Prerequisites: Programming experience and familiarity with basic discrete and numerical algorithms.* Practical aspects of application development for high performance computing. Programming language choice; compilers; compiler usage. Build management using make and other tools. Library development and usage. Portability and the GNU auto-configure system. Correctness and performance debugging, performance analysis. Group development practices and version control. Use of third-party libraries and software licensing.

CS 292 Parallel Programming Paradigms (3-0-3) *Prerequisites: Programming experience and familiarity with basic discrete and numerical algorithms.* Distributed and shared memory programming models and frameworks. Thread programming and OpenMP. Message passing and MPI. Parallel Global Address Space (PGAS) languages. Emerging languages for many core programming. Elements to be covered will include syntax and semantics, performance issues, thread safety and hybrid programming paradigms.

CS 308. Stochastic Methods in Engineering (3-0-3) *Prerequisite: CS 241.* Review of basic probability; Monte Carlo simulation; statespace models and time series; parameter estimation, prediction and filtering; Markov chains and processes; stochastic control and stochastic differential equations. Examples from various engineering disciplines.

CS 311 High Performance Computing I (3-0-3) *Prerequisite: Programming experience and familiarity with basic discrete and numerical algorithms*. Part one of a two-course sequence in high performance computing technology, with an emphasis on using KAUST's research computing systems, focusing primarily on hardware architectures. History of high performance computing. Hardware architectures. CMOS processor design. Cache architectures. Memory architectures. Hardware counters. Processing benchmarks. Power. Single-node performance of real applications.

CS 312 High Performance Computing II (3-0-3) *Prerequisites: Programming experience and familiarity with basic discrete and numerical algorithms and AMCS 311.* Part two of a two-course sequence in high performance computing technology, with an emphasis on using KAUST's research computing systems, focusing primarily on hardware architectures. I/O systems and communication networks. Communication benchmarks. Theoretical and achievable performance for processor, memory system, network, and I/O. Future architecture directions and limitations. The course is intended to develop a deep understanding of the underlying high performance computing architectures on which the student will develop and deploy applications.

CS 330. Computational Science and Engineering (3-0-3) *Prerequisite: Programming experience and familiarity with basic discrete and numerical algorithms and experience with one or more computational applications*. Case studies of representative and prototype applications in partial differential equations and mesh-based methods, particle methods, ray-tracing methods, transactional methods.

CS 337. Information Networks (3-0-3) *Prerequisite: probability. Network structure of the Internet and the Web.* Modeling, scale-free graphs, small-world phenomenon. Algorithmic implications in searching and inter-domain routing, the effect of structure on performance. Game theoretic issues, routing games and network creation games. Security issues, vulnerability and robustness.

CS 340. Computational Methods in Data Mining (3-0-3) *Prerequisites: Probability and scientific computing. Focus is on very large-scale data mining.* Topics include computational methods in supervised and unsupervised learning, association mining and collaborative filtering. Individual or group applications oriented programming project. 1 unit without project; 3 units requires final project.

CS 341 Advanced Topics in Data Management (3-0-3) *Prerequisites: CS 245 Topics in Data Management will be analyzed and discussed.* Students will engage in research and project presentations. Topics will vary by semester.

CS 344 Advanced Topics in Computer Networks (3-0-3) *Prerequisites: CS 244. Solid computer networks background, excellent skills in C/C++ and TCL, suing network simulators such as NS-2, working with Linux systems.* Topics in Computer Networks will be analyzed and discussed. Topics will vary by semester.

CS 346. Advanced Topics in Operating Systems (3-0-3) *Prerequisites: Solid computer programming skills (at least at the level of CS 207) and solid background in at least one operating systems (CS 240) or computer architecture (at least at the level of CS 209 or CS 282), or permission of instructor.* Topics in Operating Systems will be analyzed and discussed. Topics will vary by semester.

CS 380 GPU and GPGPU Programming (3-0-3) *Prerequisite: CS 280. Recommended optional prerequisites: CS 248, CS 292.* Architecture and programming of GPUs (Graphics Processing Units). Covers both the traditional use of GPUs for graphics and visualization, as well as their use for general purpose computations (GPGPU). GPU many-core hardware architecture, shading and computer programming languages and APIs, programming vertex, geometry, and fragment shaders, programming with CUDA, Brook, OpenCL, stream computing, approaches to massively parallel computations, memory subsystems and caches, rasterization, texture mapping, linear algebra computations, alternative and future architectures.

CS 361 Combinatorial Machine Learning (3-0-3) *Prerequisites: CS 260.* Lower and upper bounds on complexity and algorithms for construction (optimization) of decision trees, decision rules and tests. Decision tables with one-valued decisions and decision tables with many-valued decisions. Approximate decision trees, rules and tests. Global and local approaches to the study of problems over infinite sets of attributes. Applications to discrete optimization, fault diagnosis, pattern recognition, analysis of acyclic programs, data mining and knowledge discovery. Current results of research.

CS 297. Master Dissertation Research (variable credit) Master-level supervised research.

CS 298. Master Graduate Seminar (variable credit) Master-level seminar focusing on special topics within the field.

CS 299. Master Directed Research (variable credit) Master-level supervised research.

CS 397. Doctoral Dissertation Research (variable credit) Doctoral-level supervised research.

CS 398. Doctoral Graduate Seminar (variable credit) Doctoral-level seminar focusing on special topics within the field.

CS 399. Doctoral Directed Research (variable credit) Doctoral-level supervised research. entation, probabilistic models, machine learning, neural networks, vision, robotics and natural language understanding.

CS 229. Machine Learning (3-0-3) *Prerequisites: linear algebra and basic probability and statistics. Familiarity with artificial intelligence recommended.* Topics: statistical pattern recognition, linear and non-linear regression, nonparametric methods, exponential family, GLIMs, support vector machines, kernel methods, model/ feature selection, learning theory, VC dimension, clustering, density estimation, EM, dimensionality reduction, ICA, PCA, reinforcement learning and adaptive control, Markov decision processes, approximate dynamic programming and policy search.

CS 240. Operating Systems and Systems Programming (3-0-3) *Prerequisite: solid computer programming skills (at least at the level of CS 207).* Operating systems design and implementation. Basic structure; synchronization and communication mechanisms; implementation of processes, process management, scheduling and protection; memory organization and management, including virtual memory; I/O device management, secondary storage and file systems.

CS 241. Probability and Random Process (3-0-3) *Prerequisites: Advanced and multivariate calculus.* Introduction to probability and random processes. Topics include probability axioms, sigma algebras, random vectors, expectation, probability distributions and densities, Poisson and Wiener processes, stationary processes, autocorrelation, spectral density, effects of filtering, linear least squares estimation and convergence of random sequences.

CS 242. Programming Languages (3-0-3) *Prerequisites: programming experience with Lisp, C and an object-oriented programming language.* Central concepts in modern programming languages, impact on software development, language design trade-o s and implementation considerations. Functional, imperative and object oriented paradigms. Formal semantic methods and program analysis. Modern type systems, higher-order functions and closures, exceptions and continuations. Modularity, object-oriented languages, and concurrency. Runtime support for language features, interoperability and security issues.

CS 243. Compilers (3-0-3) *Prerequisites: solid computer programming skills (at least at the level of CS 207), familiarity with formal languages (regular expressions and grammars).* Principles and practices for design and implementation of compilers and interpreters. Topics: lexical analysis, parsing theory, symbol tables, type systems, scope, semantic analysis, intermediate representations, run-time environments, co de generation and basic program analysis and optimization. Students construct a compiler for a simple object-oriented language during course programming projects.

CS 244. Computer Networks (3-0-3) *Prerequisite: Operating systems and systems programming.* Packet switching, Internet architecture, routing, router architecture, control algorithms, retransmission algorithms, congestion control, TCP/IP, detecting and recovering from errors, switching, Ethernet (wired and wireless) and local area networks, physical layers. clocking and synchronization. Assignments intro duce network programming, including sockets, designing a router and implementing a transport layer.

CS 245. Databases (3-0-3) *Prerequisites: working knowledge of basic discrete mathematics (e.g., sets, functions and relations) and programming skills.* Database design and use of database management systems for applications. The relational model, relational algebra and SQL, the standard language for creating, querying and modifying relational and object-relational databases. XML data including the query languages XPath and XQuery. UML database design and relational design principles based on functional dependencies and normal forms. Other topics include indexes, views, transactions, authorization, integrity constraints and triggers. Advanced topics from data warehousing, data mining, Web data management, Datalog, data integration, data streams and continuous queries and data-intensive Web services.

CS 247. Scientific Visualization (3-0-3) *Prerequisites: Advanced and multivariate calculus, and linear algebra, computer graphics, and programming experience.* Techniques for generating images of various types of experimentally measured, computer generated, or gathered data. Grid structures. Scalar field visualization. Vector field visualization. Particle visualization. Graph visualization. Animation. Applications in science, engineering, and medicine.

CS 248. Computer Graphics (3-0-3) *Prerequisites: solid programming skills and linear algebra.* Input and display devices, scan conversion of geometric primitives, 2D and 3D geometric transformations, clipping and windowing, scene modeling and animation, algorithms for visible surface determination, local and global shading models, color and realtime rendering methods.

CS 251. Numerical Linear Algebra (3-0-3) *Prerequisites: Programming skills and linear algebra.* Linear algebra in the presence of floating point rounding error, with applications to large scale scientific computing. Matrix factorizations. Linear least squares. Accuracy and stability. Eigen analysis. Singular Value Decomposition. Krylov subspace methods and preconditioning. Optimization and saddle point systems.

CS 260. Design and Analysis of Algorithms (3-0-3) *Prerequisites: computer programming skills, knowledge of probability, understanding of basic data structures, basic knowledge in discrete mathematics.* Fulfills University Mathematics Requirement. Review of algorithm analysis (search in ordered array, binary insertion sort, merge sort, 2-3 trees, asymptotic notation). Divide and conquer algorithms (master theorem, integer multiplication, matrix multiplication, fast Fourier transform). Graphs (breadth-first search, connected components, topological ordering, depth-first search). Dynamic programming (chain matrix multiplication, shortest paths, edit distance, sequence alignment). Greedy algorithms (binary heaps, Dijkstra's algorithm, minimum spanning tree, Huffman codes). Randomized algorithms (selection, quick sort, global minimum cut, hushing). P and NP (Cook's theorem, examples of NP-complete problems). Approximate algorithms for NP-hard problems (set cover, vertex cover, maximum independent set). Partial recursive functions (theorem of Post, Diophantine equations). Computations and undecidable problems (undecidability of halting problem, theorem of Rice, semantic and syntactical properties of programs).

CS 261. Algorithmic Paradigms (3-0-3) *Prerequisite: Familiarity with discrete algorithms at the level of AMCS 260.* Topics: algorithms for optimization problems such as matching, max flow, min-cut and load balancing. Using linear programming, emphasis is on LP duality for design and analysis of approximation algorithms. Approximation algorithms for NP-complete problems such as Steiner trees, traveling salesman and scheduling problems. Randomized algorithms.

CS 271. Applied Geometry (3-0-3) Differential Geometry: selected topics from the classical theory of curves and surfaces, geometric variational problems, robust computation of differential invariants, discrete differential geometry. Projective Geometry: computing with homogeneous coordinates, projective maps, quadrics and polarity. Algebraic Geometry: algebraic curves and surfaces, rational parametrizations, basic elimination theory. Kinematical Geometry: geometry of motions, kinematic mappings. The practical use of these topics is illustrated at hand of sample problems from Geometric Modeling, Computer Vision, Robotics and related areas of Geometric Computing.

CS 272 Geometric Modeling (3-0-3) *Prerequisites: Advanced and multivariate calculus, and linear algebra, computer graphics, and programming experience.* Terminology, coordinate systems, and implicit forms. Parametric and spline representations of curves and surfaces and their uses. Basic differential geometry of curves and surfaces. Subdivision surfaces. Solid modeling paradigms and operations. Robustness and accuracy in geometric computations. Applications.

CS 282. Computer Architecture and Organization (3-0-3) *Prerequisite: CS 209.* Advanced topics in cache hierarchies, memory systems, storage and IO systems, interconnection networks and message passing multi-processor systems (clusters). Issues such as locality, coarse grain parallelism, synchronization, overlapping communication with computation, hardware/software interfaces, performance/power trade-o s and reliability. Characteristics of modern processors that affect system architecture.

CS 291 Scientific Software Engineering (3-0-3) *Prerequisites: Programming experience and familiarity with basic discrete and numerical algorithms.* Practical aspects of application development for high performance computing. Programming language choice; compilers; compiler usage. Build management using make and other tools. Library development and usage. Portability and the GNU auto-configure system. Correctness and performance debugging, performance analysis. Group development practices and version control. Use of third-party libraries and software licensing.

CS 292 Parallel Programming Paradigms (3-0-3) *Prerequisites: Programming experience and familiarity with basic discrete and numerical algorithms.* Distributed and shared memory programming models and frameworks. Thread programming and OpenMP. Message passing and MPI. Parallel Global Address Space (PGAS) languages. Emerging languages for many core programming. Elements to be covered will include syntax and semantics, performance issues, thread safety and hybrid programming paradigms.

CS 308. Stochastic Methods in Engineering (3-0-3) *Prerequisite: CS 241.* Review of basic probability; Monte Carlo simulation; statespace models and time series; parameter estimation, prediction and filtering; Markov chains and processes; stochastic control and stochastic differential equations. Examples from various engineering disciplines.

CS 311 High Performance Computing I (3-0-3) *Prerequisite: Programming experience and familiarity with basic discrete and numerical algorithms*. Part one of a two-course sequence in high performance computing technology, with an emphasis on using KAUST's research computing systems, focusing primarily on hardware architectures. History of high performance computing. Hardware architectures. CMOS processor design. Cache architectures. Memory architectures. Hardware counters. Processing benchmarks. Power. Single-node performance of real applications.

CS 312 High Performance Computing II (3-0-3) *Prerequisites: Programming experience and familiarity with basic discrete and numerical algorithms and AMCS 311.* Part two of a two-course sequence in high performance computing technology, with an emphasis on using KAUST's research computing systems, focusing primarily on hardware architectures. I/O systems and communication networks. Communication benchmarks. Theoretical and achievable performance for processor, memory system, network, and I/O. Future architecture directions and limitations. The course is intended to develop a deep understanding of the underlying high performance computing architectures on which the student will develop and deploy applications.

CS 330. Computational Science and Engineering (3-0-3) *Prerequisite: Programming experience and familiarity with basic discrete and numerical algorithms and experience with one or more computational applications*. Case studies of representative and prototype applications in partial differential equations and mesh-based methods, particle methods, ray-tracing methods, transactional methods.

CS 337. Information Networks (3-0-3) *Prerequisite: probability. Network structure of the Internet and the Web.* Modeling, scale-free graphs, small-world phenomenon. Algorithmic implications in searching and inter-domain routing, the effect of structure on performance. Game theoretic issues, routing games and network creation games. Security issues, vulnerability and robustness.

CS 340. Computational Methods in Data Mining (3-0-3) *Prerequisites: Probability and scientific computing. Focus is on very large-scale data mining.* Topics include computational methods in supervised and unsupervised learning, association mining and collaborative filtering. Individual or group applications oriented programming project. 1 unit without project; 3 units requires final project.
CS 341 Advanced Topics in Data Management (3-0-3) *Prerequisites: CS 245 Topics in Data Management will be analyzed and discussed.* Students will engage in research and project presentations. Topics will vary by semester.

CS 344 Advanced Topics in Computer Networks (3-0-3) *Prerequisites: CS 244. Solid computer networks background, excellent skills in C/C++ and TCL, suing network simulators such as NS-2, working with Linux systems.* Topics in Computer Networks will be analyzed and discussed. Topics will vary by semester.

CS 346. Advanced Topics in Operating Systems (3-0-3) *Prerequisites: Solid computer programming skills (at least at the level of CS 207) and solid background in at least one operating systems (CS 240) or computer architecture (at least at the level of CS 209 or CS 282), or permission of instructor.* Topics in Operating Systems will be analyzed and discussed. Topics will vary by semester.

CS 380 GPU and GPGPU Programming (3-0-3) *Prerequisite: CS 280. Recommended optional prerequisites: CS 248, CS 292.* Architecture and programming of GPUs (Graphics Processing Units). Covers both the traditional use of GPUs for graphics and visualization, as well as their use for general purpose computations (GPGPU). GPU many-core hardware architecture, shading and computer programming languages and APIs, programming vertex, geometry, and fragment shaders, programming with CUDA, Brook, OpenCL, stream computing, approaches to massively parallel computations, memory subsystems and caches, rasterization, texture mapping, linear algebra computations, alternative and future architectures.

CS 361 Combinatorial Machine Learning (3-0-3) *Prerequisites: CS 260.* Lower and upper bounds on complexity and algorithms for construction (optimization) of decision trees, decision rules and tests. Decision tables with one-valued decisions and decision tables with many-valued decisions. Approximate decision trees, rules and tests. Global and local approaches to the study of problems over infinite sets of attributes. Applications to discrete optimization, fault diagnosis, pattern recognition, analysis of acyclic programs, data mining and knowledge discovery. Current results of research.

CS 297. Master Dissertation Research (variable credit) Master-level supervised research.

CS 298. Master Graduate Seminar (variable credit) Master-level seminar focusing on special topics within the field.

CS 299. Master Directed Research (variable credit) Master-level supervised research.

CS 397. Doctoral Dissertation Research (variable credit) Doctoral-level supervised research.

CS 398. Doctoral Graduate Seminar (variable credit) Doctoral-level seminar focusing on special topics within the field.

CS 399. Doctoral Directed Research (variable credit) Doctoral-level supervised research.