EN.605 (COMPUTER SCIENCE)

EN.605.101. Introduction to Python.
Not for a letter grade. Offered pass/fail only. This is a six-week course. The withdrawal deadline is the end of the fourth week. Students must pass each module to pass the course.Course Note(s): Not for graduate credit. This course does not satisfy any admission requirements.Instructor(s): Non-facilitated

EN.605.201. Intro to Programming Using Java. 3 Credits.
This course enables students without a background in software development to become proficient programmers who are prepared for a follow-on course in data structures. The Java language will be used to introduce foundations of structured, procedural, and object-oriented programming. Topics include I/O, data types, operators, operands, expressions, conditional statements, iteration, recursion, arrays, functions, parameter passing, and returning values. Students will also be introduced to classes, objects, object references, inheritance, polymorphism, and exception handling. Additional topics include file I/O, searching, sorting, Java Collections, and an introduction to Applets. Students will complete several programming assignments to develop their problem-solving skills and to gain experience in detecting and correcting software errors. Prerequisite(s): One year of college mathematics. Course Note(s): Not for graduate credit. A programming methodology course is needed for admission to the Computer Science, Cybersecurity, Data Science, or Information Systems Engineering program. Students who lack this prerequisite can fulfill admission requirements by completing this course with a grade of B– or better.

EN.605.202. Data Structures. 3 Credits.
This course investigates abstract data types (ADTs), recursion, algorithms for searching and sorting, and basic algorithm analysis. ADTs to be covered include lists, stacks, queues, priority queues, trees, sets, and dictionaries. The emphasis is on the trade-offs associated with implementing alternative data structures for these ADTs. There will be four or five substantial Java programming assignments. (Not for Graduate credit.) Prerequisite(s): One year of college mathematics. 605.201 Introduction to Programming Using Java or equivalent. Course Note(s): Not for graduate credit. A course in data structures is needed for admission to the Computer Science, Cybersecurity, or Data Science program. Students who lack this prerequisite can fulfill admission requirements by completing this course with a grade of B– or better. A course in data structures is conditionally required for admission to the Information Systems Engineering program. Students who lack this prerequisite can satisfy it by completing this course with a grade of B– or better before taking any course that requires it.

EN.605.203. Discrete Mathematics. 3 Credits.
This course emphasizes the relationships between certain mathematical structures and various topics in computer science. Topics include set theory, graphs and trees, algorithms, propositional calculus, logic and induction, functions, relational algebra, and matrix algebra. Prerequisite(s): Calculus is recommended. Course Note(s): Not for graduate credit. A mathematics course beyond one year of calculus is needed for admission to the Computer Science, Cybersecurity, or Data Science program. A course in either calculus or discrete mathematics is needed for admission to the Information Systems Engineering program. Students who lack this prerequisite can fulfill admission requirements by completing this course with a grade of B– or better.

EN.605.204. Computer Organization. 3 Credits.
This course examines how a computer operates at the machine level. Students will develop an understanding of the hardware/ software interface by studying the design and operation of computing system components. In addition, students will program at the assembly language level to understand internal system functionality. Finally, students will become familiar with the machine representations of programs and data, as well as the influence of the underlying hardware system on the design of systems software such as operating systems, compilers, assemblers, and linkers and loaders. Prerequisite(s): 605.202 - Data Structures is recommended. Course Note(s): Not for graduate credit. A course in computer organization is needed for admission to the Computer Science or Cybersecurity program. Students who lack this prerequisite can fulfill admission requirements by completing this course with a grade of B– or better.

EN.605.205. Molecular Biology for Computer Scientists. 3 Credits.
This course is designed for students who seek to take bioinformatics courses but lack prerequisites in the biological sciences. The course covers essential aspects of biochemistry, cell biology, and molecular biology. Topics include the chemical foundations of life; cell organization and function; the structure and function of macromolecules; gene expression— transcription, translation, and regulation; biomembranes and transmembrane transport; metabolism and cellular energetics; and signal transduction. The application of foundational concepts in developmental biology, neurobiology, immunology, and cancer biology is also introduced. Course Note(s): Not for graduate credit. Several courses in the Bioinformatics track of Computer Science require background in Molecular Biology. Students can fulfill this requirement by completing this course with a grade of B- or better.

EN.605.206. Introduction to Programming Using Python. 3 Credits.
This course is a practical introduction for those interested in learning Python for a wide variety of applications and use cases. The material has been designed to expose you to common techniques and tools you'll be able to exercise immediately. This course assumes no prior development experience and ranges from beginning to intermediate Python concepts including: creating a Python environment, data types, operators/expressions, data and control structures, conditional statements, classes/objects, functions, multi-threaded applications, testing and deployment tools, REST API's, machine learning, and more. You'll also gain valuable experience with tools like PyCharm/ VSCode, Jupyter Notebooks, Git, PyLint, PyDocs/Doxygen, and many more. Each concept is accompanied by real code samples that will be explained in-detail and the assignments will present you with interesting scientific problems to enable you to practice your Python skills for the purpose of solving real, complex problems. The course is textbook-free and provides a number of hand-chosen readings to supplement the lecture materials. Upon completion of the course you will be equipped with knowledge of the skills and tools to begin tackling problems the Pythonic way.Prerequisite(s): One year of college mathematics.Course Note(s): Not for graduate credit. A programming methodology course is needed for admission to the Artificial Intelligence or Data Science. Students who lack this prerequisite can fulfill admission requirements by completing this course with a grade of B– or better

EN.605.601. Foundations of Software Engineering. 3 Credits.
Fundamental software engineering techniques and methodologies commonly used during software development are studied. Topics include various life cycle models, project planning and estimation, requirements analysis, program design, construction, testing, maintenance and implementation, software measurement, and software quality. Emphasized are structured and object-oriented analysis and design techniques, use of process and data models, modular principles of software design, and a systematic approach to testing and debugging. The importance of problem specification, programming style, periodic reviews, documentation, thorough testing, and ease of maintenance are covered. Course Note(s): The required foundation courses may be taken in any order but must be taken before other courses in the degree.

EN.605.602. Software Analysis and Design. 3 Credits.
This course prepares students to successfully engineer secure software systems by addressing critical security challenges across the entire software development life cycle. Students will learn the practical skills for building secure software from the ground up through hands-on labs and exercises. Key topical areas addressed include security in software requirements, design, and development. Common security pitfalls are highlighted and examined as well as the tools and techniques for identifying and eliminating the security vulnerabilities. Security considerations in Mobile code development are also addressed. Parameterized refinement methods and transduction techniques based on mathematical-based proofs are presented as a means of verifying the correctness of code and modifications to code as well as to validate conformance with functional requirements. Software protection techniques such as code obfuscation and water-marking are explored.

EN.605.603. Object-Oriented and Functional Programming in Kotlin. 3 Credits.
This course introduces object-oriented and functional programming in the new programming language Kotlin. Kotlin runs on multiple platforms and virtually anywhere, compiling to native code, JavaScript, the Android runtime, and the Java Virtual Machine. It easily interacts with other Java code. Through this course, you'll become adept at Kotlin programming, an easier-to-use, safer and more productive language than Java. We'll cover the basics of the language, including data types, functions and collections, object-oriented features such as classes, encapsulation, inheritance, composition, delegation and generics, and functional features such as immutability, higher-order functions and functional chaining. You'll learn how to create multi-threaded applications using coroutines and builders that will simplify the use of your libraries using simple Domain-Specific languages. Students will build several projects in Kotlin.Pre-requisites: Competence in a procedural language (such as C, Pascal, or Visual Basic) or object-oriented language (such as Java or C++). Note that this is not an "introduction to programming" class and cannot substitute for 605.201; we assume familiarity with programming in general.

EN.605.604. Object-Oriented Programming with C++. 3 Credits.
This course provides in-depth coverage of object-oriented programming principles and techniques using C++. Topics include classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, file processing, templates, exceptions, container classes, and lowlevel language features. The course briefly covers the mapping of UML design to C++ implementation and object-oriented considerations for software design and reuse. The course also relates C++ to GUI, databases, and real-time programming. The course material embraces the C++11 language standard with numerous examples demonstrating the benefits of C++11. Prerequisite(s): Knowledge of Java or C++.

EN.605.606. Programming with Domain-Specific Languages. 3 Credits.
Domain-specific languages (DSLs) are little languages you write that look and feel like a spoken way to specify data or write code. You can use them for input and output, incorporating the jargon and nomenclature of your subject-matter experts (SMEs), as well as inside your own code to make it more expressive and fluent, and often simpler. You can use them as part of your build process to generate hundreds of classes full of otherwise tedious and error-prone boilerplate code from a small specification in a consistent manner. In this course, we'll design and implement several types of DSLs. We'll write code to edit and import data, allowing SMEs more natural-feeling access to your software. We'll create APIs in multiple programming languages to make it easier and more secure for others to use your libraries. We'll generate code to improve productivity and reliability in your own software. Course Note(s): Examples and assignments in this class will be done in several programming languages. We assume a high comfort level with Java and the ability to adapt to new languages quickly.

EN.605.607. Agile Software Development Methods. 3 Credits.
This course emphasizes the quick realization of system value through disciplined, iterative, and incremental software development techniques and the elimination of wasteful practices. Students will study the full spectrum of agile methods, including Scrum, Extreme Programming, Lean, Kanban, Dynamic Systems Development Method, and Feature-Driven Development. These methods promote teamwork, rich concise communication, and the frequent delivery of running, tested systems containing the highest-priority stakeholder features. Agile methods are contrasted with common workplace practices and traditional methods such as Waterfall, CMMI, and PMI/ PMBOK. Examples of agile adoption in industry are discussed. Assignments and projects are designed to help students apply agile principles and practices in their own professional context. Additional subthemes in the course include enterprise agility, team dynamics, collaboration, software quality, and metrics for reporting progress. Prerequisite(s): 605.202 Data Structures.

EN.605.608. Software Project Management. 3 Credits.
This course describes the key aspects of a software project. It begins with the job description of a software manager and then addresses those topics germane to successful software development management, including organizing the software development team; interfacing with other engineering organizations (systems engineering, quality assurance, configuration management, and test engineering); assessing development standards; selecting the best approach and tailoring the process model; estimating software cost and schedule; planning and documenting the plan; staffing the effort; managing software cost and schedule during development; risk engineering; and continuous process improvement. Personnel management topics, including performance evaluations, merit planning, skills building, and team building, are also covered. This course introduces software engineers aspiring to become technical team leaders or software project managers to the responsibilities of these roles. For those engineers who have advanced to a software development leadership position, this course offers formal training in software project management. Prerequisite(s): Three to five years technical work experience is recommended.

EN.605.609. DevOps Software Development. 3 Credits.
"DevOps" evokes an agile software development approach in an operational environment. Modern technologies, particularly cloud and big data analytics, often embrace DevOps methods.The term was first used to indicate "agile infrastructure." Recently, it has often referred to quick adoption of emerging technology and subsequent integration into production. This course gathers the latest publications to examine the tools for source code control, automated build, automated test, and automated deployment, some of which will be selected by the students in an operational rhythm of Continuous Integration (CI) and Continuous Deployment (CD). This course discusses the basic concepts of DevOps, including its philosophy, workflow, monitoring methods, and tools. Topics include: concepts and vision for DevOps, release/deployment pipelines, automated testing, monitoring production, task evaluation, skills assessment, and tool selection. Students will apply these concepts to see how they can be best implemented to automate development, test, and release practices. Students will work in teams to build functional working models of realized DevOps. Prerequisite(s): Prior experience in software development in any language is required. Familiarity with software design, development, and architecture techniques is recommended.

EN.605.611. Foundations of Computer Architecture. 3 Credits.
This course provides a detailed examination of the internal structure and operation of modern computer systems. Each of the major system components is investigated, including the following topics: the design and operation of the ALU, FPU, and CPU; microprogrammed vs. hardwired control, pipelining, and RISC vs. CISC machines; the memory system including caches and virtual memory; parallel and vector processing, multiprocessor systems and interconnection networks; superscalar and super-pipelined designs; and bus structures and the details of low-level I/O operation using interrupt mechanisms, device controllers, and DMA. The impact of each of these topics on system performance is also discussed. The instruction set architectures and hardware system architectures of different machines are examined and compared. The classical Von Neumann architecture is also compared and contrasted with alternative approaches such as data flow machines and neural networks. Course Note(s): The required foundation courses may be taken in any order but must be taken before other courses in the degree.

EN.605.612. Operating Systems. 3 Credits.
The theory and concepts related to operating system design are presented from both developer and user perspectives. Core concepts covered include process management, memory management, file systems, I/O system management including device drivers, distributed systems, and multi-user concepts including protection and security. Process management discussions focus on threads, scheduling, and synchronization. Memory management topics include paging, segmentation, and virtual memory. Students will examine how these concepts are realized in several current open-source operating systems, including Linux. Students will complete several assignments that require the design and implementation of operating system programs using a high-level language.

EN.605.613. Introduction to Robotics. 3 Credits.
This course introduces the fundamentals of robot design and development with an emphasis on autonomy. Robot design, navigation, obstacle avoidance, and artificial intelligence will be discussed. Topics covered in robot design include robot structure, kinematics and dynamics, the mathematics of robot control (multiple coordinate systems and transformations), and designing for autonomy. Navigation topics include path planning, position estimation, sensors (e.g., vision, ultrasonics, and lasers), and sensor fusion. Obstacle avoidance topics include obstacle characterization, object detection, sensors and sensor fusion. Topics to be discussed in artificial intelligence include learning, reasoning, and decision making. Students will deepen their understanding through several assignments and the term-long robot development project.

EN.605.614. System Development in the UNIX Environment. 3 Credits.
This course describes how to implement software systems in a UNIX (POSIX-compliant) operating system environment. Students will discuss and learn the complexities, methodologies, and tools in the development of large systems that contain multiple programs. Topics include an overview of the UNIX system and its general-purpose tools, advanced makefile usage, UNIX system calls, UNIX process management, threads, and basic and advanced interprocess communication. Additional topics include source code configuration control, Perl, and debugging techniques.Prerequisite(s): Familiarity with UNIX, experience with C++ or C.

EN.605.615. Compiler Design with LLVM. 3 Credits.
The components of a compiler appear in every software application that handles input from an external source. This course shows how the components of a compiler are built and how they fit together to extract meaning from the input and how the data flows through the compiler's components to become useful to applications. Students will get practical experience in how to use the LLVM tools to build a complete compiler for a subset of the C++ programming language that can target almost any platform. Students will also get experience in developing a "Just In Time" component for an application that will accept code as input into the application while it is running, to be compiled and linked into the application so the application can execute it.

EN.605.616. Multiprocessor Architecture & Programming. 3 Credits.
This course addresses how to utilize the increasing hardware capabilities of multiprocessor computer architecture's highperformance computing platforms for software development. The famous Moore's Law is still alive, although it is now realized from increasing the number of CPU cores instead of increasing CPU clock speed. This course describes the differences between single-core and multi-core systems and addresses the impact of these differences in multiprocessor computer architectures and operating systems. Parallel programming techniques to increase program performance by leveraging the multiprocessor system, including multi-core architectures, will be introduced. Additional topics include program performance analysis and tuning, task parallelism, synchronization strategies, shared memory access and data structures, and task partition techniques. The course encourages hands-on experience with projects selected by the student.

EN.605.617. Introduction to GPU Programming. 3 Credits.
This course will teach the fundamentals needed to utilize the ever-increasing power of the GPUs housed in the video cards attached to our computers. For years, this capability was limited to the processing of graphics data for presentation to the user. With the CUDA and OpenCL frameworks, programmers can develop applications that harness this power directly to search, modify, and quickly analyze large amounts of various types of data. Students will be introduced to core concurrent programming principles, along with the specific hardware and software considerations of these frameworks. In addition, students will learn canonical algorithms used to perform high-precision mathematics and data transformations. Class time will be split between lectures and hands-on exercises. There will be two individual projects in both CUDA and OpenCL programming, which will allow students to independently choose demonstrable goals, develop software to achieve those goals, and present the results of their efforts.

EN.605.620. Algorithms for Bioinformatics. 3 Credits.
This follow-on course to data structures (e.g., 605.202 Data Structures) provides a survey of computer algorithms, examines fundamental techniques in algorithm design and analysis, and develops problem-solving skills required in all programs of study involving computer science. Topics include advanced data structures (red-black and 2-3-4 trees, union-find), algorithm analysis and computational complexity (recurrence relations, big-O notation, introduction to NP-completeness), sorting and searching, design paradigms (divide and conquer, greedy heuristic, dynamic programming), and graph algorithms (depth-first and breadth-first search, minimum spanning trees). Advanced topics are selected from among the following: multithreaded algorithms, matrix operations, linear programming, string matching, computational geometry, and approximation algorithms. The course will draw on applications from Bioinformatics. Prerequisite(s): 605.202 Data Structures or equivalent, and 605.201 Introduction to Programming Using Java or equivalent. 605.203 Discrete Mathematics or equivalent is recommended.Course Note(s): This course does not satisfy the foundation course requirement for Data Science, Computer Science, or Cybersecurity. Students can only earn credit for one of 605.620, 605.621, or 685.621.
Prerequisite(s): ;

EN.605.621. Foundations of Algorithms. 3 Credits.
This follow-on course to data structures (e.g., 605.202) provides a survey of computer algorithms, examines fundamental techniques in algorithm design and analysis, and develops problem-solving skills required in all programs of study involving computer science. Topics include advanced data structures (red-black and 2-3-4 trees, union-find), recursion and mathematical induction, algorithm analysis and computational complexity (recurrence relations, big-O notation, NP-completeness), sorting and searching, design paradigms (divide and conquer, greedy heuristic, dynamic programming, amortized analysis), and graph algorithms (depth-first and breadth-first search, connectivity, minimum spanning trees, network flow). Advanced topics are selected from among the following: randomized algorithms, information retrieval, string and pattern matching, and computational geometry. Prerequisite(s): 605.202 Data Structures or equivalent. 605.203 Discrete Mathematics or equivalent is recommended. Course Note(s): The required foundation courses may be taken in any order but must be taken before other courses in the degree. Students can only earn credit for one of 605.620, 605.621, or 685.621.
Prerequisite(s): ;

EN.605.622. Computational Signal Processing. 3 Credits.
This course introduces computational aspects of signal processing, specifically algorithms for processing digital signals, methods for the design and analysis of signal processing algorithms, architectures for signal processing systems, and areas of application. Topics include signal analysis (signal definition, time and frequency domains, sampling and digitizing, noise and error), systems for signal processing (filters and nonfilters, correlation, adaptation), and algorithms and architectures (fast Fourier transforms, fast convolution, digital filtering, interpolation and resampling, digital signal processors, function evaluation, and computational complexity). Areas of application include communication systems, speech signal processing, and digital media. Prerequisite(s): Knowledge of complex numbers and linear algebra.

EN.605.623. Intro to Enumerative Combinatorics. 3 Credits.
The most basic question in mathematics is How many? Counting problems arise in diverse areas including discrete probability and the analysis of the run time of algorithms.In this course we present methods for answering enumeration questions exactly and approximately. Topics include fundamental counting problems (lists, sets, partitions, and so forth), combinatorial proof, inclusion-exclusion, ordinary and exponential generating functions, group-theory methods, and asymptotics. Examples are drawn from areas such as graph theory and block designs.After completing this course students will be practiced in applying the fundamental functions (such as factorial, binomial coefficients, Stirling numbers) and techniques for solving a wide variety of exact enumeration problems as well as notation and methods for approximate counting (asymptotic equality, big-oh and little-oh notation, etc.).Course prerequisite(s): Linear algebraCourse note(s): This course is the same as 625.617 Introduction to Enumerative Combinatorics.

EN.605.624. Logic: Systems, Semantics, and Models. 3 Credits.
The use of predicate logic for modeling information systems is widespread and growing. Knowledge representation, for example, has long been important in artificial intelligence applications and is now emerging as a critical component of semantic web applications. Similarly, predicate logic is the basis for ontologies and inferential knowledge bases that support systems managing “big data” using graph databases and triple stores. This course teaches the fundamentals of propositional and predicate logic, with an emphasis on semantics. We start with a fast-paced introduction or a refresher on propositional and predicate logic, to serve as a stepping stone to more advanced topics in logics with application to computer science. Modal logic is introduced as a tool to manage non-truth functional systems, and dynamic logic is introduced to manage potentially inconsistent systems, such as may arise in merging disparate databases or in combining diagnostic models of related systems (e.g., “Agent A knows that Agent B knows fact X”), and has been key to the development of IBM’s Watson and RDF/OWL. Finally, dynamic logic is introduced to manage potentially inconsistent systems, such as may arise in merging disparate databases or in combining diagnostic models of related systems. Course Note(s): This course may be counted toward a threecourse track in Database Systems and Knowledge Management.

EN.605.625. Probabilistic Graphical Models. 3 Credits.
This course introduces the fundamentals behind the mathematical and logical framework of graphical models. These models are used in many areas of machine learning and arise in numerous challenging and intriguing problems in data analysis, mathematics, and computer science. For example, the "big data" world frequently uses graphical models to solve problems. While the framework introduced in this course will be largely mathematical, we will also present algorithms and connections to problem domains. The course will begin with the fundamentals of probability theory and will then move into Bayesian networks, undirected graphical models, templatebased models, and Gaussian networks. The nature of inference and learning on the graphical structures will be covered, with explorations of complexity, conditioning, clique trees, and optimization. The course will use weekly problem sets and a term project to encourage mastery of the fundamentals of this emerging area. Prerequisite(s): Graduate course in probability and statistics (such as 625.603 Statistical Methods and Data Analysis). Course Note(s): This course is the same as 625.692 Probabilistic Graphical Models.

EN.605.626. Image Processing. 3 Credits.
Fundamentals of image processing are covered, with an emphasis on digital techniques. Topics include digitization, enhancement, segmentation, the Fourier transform, filtering, restoration, reconstruction from projections, and image analysis including computer vision. Concepts are illustrated by laboratory sessions in which these techniques are applied to practical situations, including examples from biomedical image processing. Prerequisite(s): Familiarity with Fourier transforms.

EN.605.627. Computational Photography. 3 Credits.
Computational photography is an emerging research area at the intersection of computer graphics, image processing, and computer vision. As digital cameras become more popular and collections of images continue to grow, we've seen a surge in interest in effective ways to enhance photography and produce more realistic images through the use of computational techniques. Computational photography overcomes the limitations of conventional photography by analyzing, manipulating, combining, searching, and synthesizing images to produce more compelling, rich, and vivid visual representations of the world. This course will introduce the fundamental concepts of image processing, computer vision, and computer graphics, as well as their applications to photography. Topics include image formation, filtering, blending, and completion techniques. In addition, the course will discuss different image analysis and rendering techniques including texture analysis, morphing, and nonphotorealistic rendering.

EN.605.628. Applied Topology. 3 Credits.
The course is both an introduction to topology and an investigation of various applications of topology in science and engineering. Topology, simply put, is a mathematical study of shapes, and it often turns out that just knowing a rough shape of an object (whether that object is as concrete as platonic solids or as abstract as the space of all paths in large complex networks) can enhance one's understanding of the object. We will start with a few key theoretical concepts from point-set topology with proofs, while letting breadth take precedence over depth, and then introduce key concepts from algebraic topology, which attempts to use algebraic concepts, mostly group theory, to develop ideas of homotopy, homology, and cohomology, which render topology "computable." Finally, we discuss a few key examples of real-world applications of computational topology, an emerging field devoted to the study of efficient algorithms for topological problems, especially those arising in science and engineering, which builds upon classical results from algebraic topology as well as algorithmic tools from computational geometry and other areas of computer science. The questions we like to ask are: Do I know the topology of my network? What is a rough shape of the large data set that I am working with (is there a logical gap?) Will the local picture of a part of the sensor field I am looking at give rise to a consistent global common picture? Prerequisite(s): Multivariate calculus, linear algebra and matrix theory (e.g., Matrix Theory 625.609), and an undergraduatelevel course in probability and statistics. Course Note(s): This course is the same as 625.687 Applied Topology.

EN.605.629. Programming Languages. 3 Credits.
This course compares and contrasts a wide variety of features of at least twelve programming languages, including programming language history; formal methods of describing syntax and semantics; names, binding, type checking, and scopes; data types; expressions and assignment statements; statement-level control structures; design and implementation of subprograms; exception handling; and support for objectoriented programming. Students will also learn and write fourweek projects in three programming languages (e.g., Python, Perl, and C#).

EN.605.631. Statistical Methods for Computer Science. 3 Credits.
Statistical methods are the foundation for data science, artificial intelligence, and much of the field of computer science. Topics include probability, random variables, regression, gradient search, Bayesian methods, graphical methods, and exponential random graph models. Student will have the foundation to excel in future courses in machine learning, data science, algorithms, and more. Practice exercises will develop proficiency in the R programming language.

EN.605.632. Graph Analytics. 3 Credits.
Graphs are a flexible data structure that facilitates fusion of disparate data sets. Applications of graphs have shown steady growth with the development of Internet, cyber, and social networks, presenting large graphs for which analysis remains a challenging problem. This course introduces algorithms and techniques to address large-scale graph analytics. It will blend graph analytics theory, hands-on development of graph analytics algorithms, as well as processing approaches that support the analytics. We will start by introducing graphs, their properties, and example applications, including necessary background on probability and linear algebra. Statistical properties of random and scale-free graphs will be introduced. Graph analytic methods, including centrality measures, graph clustering, partitioning, link inference, and dynamic graph processes such as diffusion, contagion, and opinion formation will be covered. Application of graph analytics to high-dimensional data analysis and data clustering will be discussed. Students will use standard graph interfaces as well as linear algebra-based methods to analyze graphs. Parallelization of analytics to handle larger-scale graphs will be discussed. Students will identify and apply suitable algorithms and analysis techniques to a variety of graph analytics problems, as well as gain experience setting up and solving these problems. There will be hands-on programming assignments.

EN.605.633. Social Media Analytics. 3 Credits.
Today an immense social media landscape is being fueled by new applications, growth of devices (e.g., Smartphones and devices), and human appetite for online engagement. With a myriad of applications and users, significant interest exists in the obvious question, "How does one better understand human behavior in these communities to improve the design and monitoring of these communities?" To address this question a multidisciplinary approach that combines social network analysis (SNA), natural language processing, and data analytics is required. This course combines all these topics to address contemporary topics such as marketing, population influence, etc. There will be several small projects. Prerequisite(s): Knowledge of Python or R; matrix algebra.

EN.605.634. Crowdsourcing and Human Computation. 3 Credits.
Crowdsourcing and human computation reverses the typical approach to computing. Rather than using computers to conduct computation that is too difficult for a human, many humans are used to conduct computation that is too difficult for a computer. This course explores computer science topics that lie at the intersection of data science and social psychology. Topics include crowdsourcing, social media, social network analysis, games, gamification, ubiquitous computing, and computersupported cooperative work. Laboratory exercises will involve hands-on data collection and analysis to include Mechanical Turk and require programming in R or Python depending on student preference/ proficiency.

EN.605.635. Cloud Computing. 3 Credits.
Cloud computing helps organizations realize cost savings and efficiencies without spending capital resources up front, while modernizing and expanding their IT capabilities. Cloud-based infrastructure is rapidly scalable, secure, and accessible over the Internet —you pay only for what you use. So, enterprises worldwide, big and small, are moving toward cloud-computing solutions for meeting their computing needs, including the use of Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). We have also seen a fundamental shift from shrinkwrapped software to Software as a Service (SaaS) in data centers across the globe. Moreover, providers such as Amazon, Google, and Microsoft have opened their datacenters to third parties by providing low-level services such as storage, computation, and bandwidth. This trend is creating the need for a new kind of enterprise architect, developer, QA, and operational professional—someone who understands and can effectively use cloud-computing technologies and solutions. In this course, we discuss critical cloud topics such as cloud service models (IaaS, PaaS, SaaS); virtualization and how it relates to cloud; elastic computing; cloud storage; cloud networking; cloud databases; cloud security; and architecting, developing, and deploying apps in the cloud. The format of this course will be a mix of lectures, and hands-on demos. Upon completing this course, students will have a deeper understanding of what cloud computing is and the various technologies that make up cloud computing, along with hands-on experience working with a major cloud provider. Prerequisite(s): 605.202 Data Structures.

EN.605.636. Autonomic Computing. 3 Credits.
This course is an introduction to autonomic and self-aware computing systems. It surveys the field of autonomic computing from its first introductory vision to the current time. The course describes autonomic computing and how it provides self-managing systems with their ability to adapt to unpredictable changes in an environment. It concentrates on the self-awareness properties of autonomic systems, the architecture, the monitoring systems that provide the self-awareness, and the adaptation and decision making needed to adapt to changing environments. The course covers the vision of autonomic computing and how autonomic computing differs from automated and autonomous systems. It discusses the self-awareness properties of autonomic systems and their biological inspiration. Architectures of autonomic systems are covered, which includes autonomic managers that are the core of autonomic systems that provide the self-managing nature of autonomic systems. Adaptation, another important aspect of autonomic computing, is discussed as well as what makes an autonomic system self-aware. The course ends with evaluation and assurance of autonomic systems, and future trends in the field. There will be weekly readings and discussions, approximately bi-weekly assignments that go into depth on selected topics, and a final project or research paper. The project can be an implementation of a part of an autonomic computing system, or a research paper that goes into depth on one of the topics covered or a topic that is of interest to the student.

EN.605.641. Principles of Database Systems. 3 Credits.
This course examines the underlying concepts and theory of database management systems. Topics include database system architectures, transaction management, data models, query languages, conceptual and logical database design, and physical organization. The entity-relationship (ER) model, using ER diagram (ERD) and Enhanced ERD, as well as relational models, are investigated in detail. Object-oriented databases are introduced along with legacy systems based on the network. Hierarchical models as well as big data and NoSQL are also briefly described. Mappings from the conceptual level to the logical level, integrity constraints, dependencies, and normalization are studied as a basis for formal design. Theoretical languages such as the relational algebra and the relational calculus are described, and high-level languages such as SQL, triggers and Stored Procedures are discussed. An overview of file organization and access methods is provided as a basis for discussion of query optimization and execution. The course also covers the causes of performance problems and how to improve database application performance during database design and implementation.Course prerequisite(s): 605.202 Data Structures.

EN.605.643. Linked Data and the Semantic Web. 3 Credits.
The World Wide Web Consortium (W3C) is endeavoring to create standards and technology that support a distributed "Web of data." Collectively, these advances allow the systems we develop to work and interact more effectively, through the use of XML-based languages, and information on how various tags relate to real-world objects and concepts. This course covers a range of Semantic Web technologies, including RDF (Resource Description Framework - a model for data interchange) and OWL (Web Ontology Language), as well as domain-specific standards and ontologies (formal specifications of how to represent objects and concepts). Representative applications of RDF, OWL, and ontologies to various problems will be discussed. Students will apply course concepts to an in-depth project in an area of personal or professional interest. Prerequisite(s): 605.202 Data Structures. Course Note(s): This course may be counted toward a threecourse track in Bioinformatics.

EN.605.644. XML Design Paradigms. 3 Credits.
The course explores understanding the tradeoffs among XML grammars and XML techniques to solve different classes of problems. Topics include optimization of XML grammars for different XML technologies; benefits of using different XML schema languages; tradeoffs in using different parsing approaches; benefits of parsing technology vs. XML query; the role of Web 2.0 to deliver functionality through various web services approaches; exploiting XML to drive audio, visual, and tactile displays; the role of XML in multiplying the power of standard web browser technologies; and the role of Web 3.0 to deliver Semantic Web functionality. XML technologies that will be covered include XML Schema, XPath, XSLT, SAX, DOM, XQuery, SOAP, WSDL, JAX-B, JAX-WS, REST, RDF, and OWL.
Prerequisite(s): 605.681 Principles of Enterprise Web Development or equivalent Java experience.

EN.605.645. Artificial Intelligence. 3 Credits.
The incorporation of advanced techniques in reasoning and problem solving into modern, complex systems has become pervasive. Often, these techniques fall within the realm of artificial intelligence. This course focuses on artificial intelligence from an agent perspective and explores issues of knowledge representation and reasoning. Students will investigate a wide variety of approaches to artificial intelligence including heuristic and stochastic search, logical and probabilistic reasoning, planning, learning, and perception. Advanced topics will be selected from areas such as robotics, vision, natural language processing, and philosophy of mind. Students will have the opportunity to explore both the philosophical and practical issues of artificial intelligence during the course of the class.

EN.605.646. Natural Language Processing. 3 Credits.
This course surveys the principal difficulties of working with written language data, the fundamental techniques that are used in processing natural language, and the core applications of NLP technology. Topics covered in the course include language modeling, text classification, labeling sequential data (tagging), parsing, information extraction, question answering, machine translation, and semantics. The dominant paradigm in contemporary NLP uses supervised machine learning to train models based on either probability theory or deep neural networks. Both formalisms will be covered. A practical approach is emphasized in the course, and students will write programs and use open source toolkits to solve a variety of problems.Course prerequisite(s): There are no formal prerequisite courses, although having taken any of 605.649 Introduction to Machine Learning, 605.744 Information Retrieval, or 605.645 Artificial Intelligence is helpful.Course note(s): A working knowledge of Python is assumed. While some of the assigned exercises can be done in any programming language, we will sometimes provide example code in Python, and many of the labs are best solved in Python.

EN.605.647. Neural Networks. 3 Credits.
This course provides an introduction to concepts in neural networks and connectionist models. Topics include parallel distributed processing, learning algorithms, and applications. Specific networks discussed include Hopfield networks, bidirectional associative memories, perceptrons, feedforward networks with back propagation, and competitive learning networks, including self-organizing and Grossberg networks. Software for some networks is provided. Prerequisite(s): Multivariate calculus and linear algebra. Course Note(s): This course is the same as 625.638 Neural Networks.

EN.605.649. Introduction to Machine Learning. 3 Credits.
Analyzing large data sets ("Big Data"), is an increasingly important skill set. One of the disciplines being relied upon for such analysis is machine learning. In this course, we will approach machine learning from a practitioner's perspective. We will examine the issues that impact our ability to learn good models (e.g., the curse of dimensionality, the biasvariance dilemma, and no free lunch). We will then examine a variety of approaches to learning models, covering the spectrum from unsupervised to supervised learning, as well as parametric versus non-parametric methods. Students will explore and implement several learning methods, including logistic regression, Bayesian classification, decision trees, and feed-forward neural networks, and will incorporate strategies for addressing the issues impacting performance (e.g., regularization, clustering, and dimensionality reduction). In addition, students will engage in online discussions, focusing on the key questions in developing learning systems. At the end of this course, students will be able to implement and apply a variety of machine learning methods to real-world problems, as well as be able to assess the performance of these algorithms on different types of data sets.

EN.605.651. Principles of Bioinformatics. 3 Credits.
This course is an interdisciplinary introduction to computational methods used to solve important problems in DNA and protein sequence analysis. The course focuses on algorithms but includes material to provide the necessary biological background for science and engineering students. Algorithms to be covered include dynamic programming for sequence alignment, such as Smith-Waterman, FASTA, and BLAST; hidden Markov models, such as the forward, Viterbi, and expectation maximization algorithms; a range of gene-finding algorithms; phylogenic tree construction; and clustering algorithms. Prerequisite(s): Familiarity with probability and statistics; working knowledge of Java, C++, C, Perl, MATLAB or Python; 605.205 Molecular Biology for Computer Scientists or a course in molecular biology; and a course in either cell biology or biochemistry.

EN.605.652. Biological Databases and Database Tools. 3 Credits.
The sequencing of thousands of genomes, including those related to disease states, interest in proteomics, epigenetics, and variation have resulted in an explosive growth in the number of biological databases, as well as the need to develop new databases to handle the diverse new content being generated. The course focuses on the design of biological databases and examines issues such as those related to data modeling, heterogeneity, interoperability, evidence, and tool integration. It also surveys a wide range of biological databases and their access tools and enables students to develop proficiency in their use. Databases introduced include genome and sequence databases such as GenBank and Ensembl, as well as protein databases such as PDB and UniProt. Databases related to RNA, sequence variation, pathways and interactions, metagenomics, and epigenomics are also presented. Tools for accessing and manipulating data from databases such as BLAST, genome browsers, multiple sequence alignment, gene finding, and protein tools are reviewed. The programming language Perl is introduced, along with the use of Perl in obtaining data via web services and in storing data in an SQLite database. Students will use Perl, biological databases, and database tools to complete homework assignments. They will also design a database and will write code in the language of their choice to create their own database as a course project.
Prerequisite(s): (For JHEP Students) 605.205 Molecular Biology for Computer Scientists or 410.634 Practical Computer Concepts for Bioinformatics or equivalent; 605.641 Principles of Database Systems or equivalent; 605.202 Data Structures and 605.201 Introduction to Programming Using Java.

EN.605.653. Computational Genomics. 3 Credits.
This course focuses on current problems of computational genomics. Students will explore bioinformatics software, discuss bioinformatics research, and learn the principles underlying a variety of bioinformatics algorithms. The emphasis is on algorithms that use probabilistic and statistical approaches. Topics include analyzing eukaryotic, bacterial, and viral genes and genomes, genome sequencing and assembling, finding genes in genomes and identifying their biological functions, predicting regulatory sites, and assessing gene and genome evolution. Prerequisite(s): 605.205 Molecular Biology for Computer Scientists or equivalent and familiarity with probability and statistics.

EN.605.656. Computational Drug Discovery,Dev. 3 Credits.
Recent advances in bioinformatics and drug discovery platforms have brought us significantly closer to the realization of rational drug design and development. Across the pharmaceutical industry, considerable effort is being invested in developing experimental and translational medicine, and it is starting to make a significant impact on the drug discovery process itself. This course examines the major steps of the evolving modern drug discovery platforms, the computational techniques and tools used during each step of rational drug discovery, and how these techniques facilitate the integration of experimental and translation medicine with the discovery/development platforms. The course will build on concepts from a number of areas including bioinformatics, computational genomic/ proteomics, in-silico system biology, computational medicinal chemistry, and pharmaceutical biotechnology. Topics covered in the course include comparative pharmacogenomics, protein/ antibody modeling, interaction and regulatory networks, QSAR/ pharmacophores, ADME/toxicology and clinical biomarkers. Relevant mathematical concepts are developed as needed in the course. Prerequisite(s): 605.205 Molecular Biology for Computer Scientists or equivalent.

EN.605.657. Statistics for Bioinformatics. 3 Credits.
This course provides an introduction to the statistical methods commonly used in bioinformatics and biological research. The course briefly reviews basic probability and statistics including events, conditional probabilities, Bayes; theorem, random variables, probability distributions, and hypothesis testing and then proceeds to topics more specific to bioinformatics research, including Markov chains, hidden Markov models, Bayesian statistics, and Bayesian networks. Students will learn the principles behind these statistical methods and how they can be applied to analyze biological sequences and data. Prerequisite(s): 605.205 Molecular Biology for Computer Scientists or equivalent, and 410.645 Biostatistics or another statistics course.

EN.605.662. Data Visualization. 3 Credits.
This course explores the underlying theory and practical concepts in creating visual representations of large amounts of data. It covers the core topics in data visualization: data representation, visualization toolkits, scientific visualization, medical visualization, information visualization, flow visualization, and volume rendering techniques. The related topics of applied human perception and advanced display devices are also introduced. Prerequisite(s): Experience with data collection/ analysis in data-intensive fields or background in computer graphics (e.g., 605.667 Computer Graphics) is recommended.

EN.605.667. Computer Graphics. 3 Credits.
This course examines the principles of computer graphics, with a focus on the mathematics and theory behind 2D and 3D graphics rendering. Topics include graphics display devices, graphics primitives, 2D and 3D transformations, viewing and projection, color theory, visible surface detection and hidden surface removal, lighting and shading, and object definition and storage methods. Practical application of these concepts is emphasized through laboratory exercises and code examples. Laboratory exercises use the C++ programming language and OpenGL on a PC. Prerequisite(s): Familiarity with linear algebra.

EN.605.671. Principles of Data Communications Networks. 3 Credits.
This course provides an introduction to the field of data communications and computer networks. The course covers the principles of data communications, the fundamentals of signaling, basic transmission concepts, transmission media, circuit control, line sharing techniques, physical and data link layer protocols, error detection and correction, data compression, common carrier services and data networks, and the mathematical techniques used for network design and performance analysis. Potential topics include analog and digital signaling; data encoding and modulation; Shannon channel capacity; synchronous and asynchronously transmission; RS232 physical layer interface standards; FDM, TDM, and STDM multiplexing techniques; inverse multiplexing; analog and digital transmission; V series modem standards; PCM encoding and T1 transmission circuits; LRC, VRC, and CRC error detection techniques; Hamming and Viterbi forward error correction techniques; BSC and HDLC data link layer protocols; Huffman, MNP5, and V.42bis data compression algorithms; circuit, message, packet, and cell switching techniques; public key and symmetric encryption algorithms, authentication, digital signature, and message digest techniques, secure e-mail, PGP, and TSL/SSL security algorithms; Ethernet, Wi-Fi, Optical, and IP networks; reliability and availability; and queuing analysis network performance techniques.

EN.605.672. Computer Network Architectures and Protocols. 3 Credits.
This course provides a detailed examination of the conceptual framework for modeling communications between processes residing on independent hosts, as well as the rules and procedures that mediate the exchange of information between two communication processes. The Open Systems Interconnection Reference Model (OSIRM) is presented and compared with TCP/IP and other network architectures. The service definitions and protocols for implementing each of the seven layers of the Reference Model using both OSI and TCP/ IP protocols are analyzed in detail. Internetworking among heterogeneous subnets is described in terms of addressing and routing, and techniques for identifying different protocol suites sent over the subnets are explained. The protocol header encoding rules are examined, and techniques for parsing protocol headers are analyzed. The application layer sub-architecture for providing common application services is described, and interoperability techniques for implementing multiprotocol internets are presented. Topics include layering, encapsulation, SAPs, and PDUs; sliding window protocols, flow and error control; virtual circuits and datagrams; routing and congestion control algorithms; internetworking; NSAP and IP addressing schemes; CLNP, IPv4, and the new IPv6 internet protocols; RIP, OSPF, ES-IS, and IS-IS routing protocols; TP4 and TCP transport protocols; dialog control, activity management, and the session layer protocol; ASN.1 encoding rules and the presentation layer protocol; application layer structure and the ACSE, CCR, ROSE, and RTSE common application service elements; OSI VT, FTAM, and MOTIS application protocols; TCP/ IP TELNET, FTP, and SMTP application protocols; OSI transitioning tools; multiprotocol networks; and encapsulation, tunneling, and convergence techniques.
Prerequisite(s): 605.671 Principles of Data Communications Networks.

EN.605.673. High-Speed Networking Technologies. 3 Credits.
The Internet has experienced unprecedented growth especially since the 1990s and is continuing to evolve in terms of the information transfer speeds and infrastructure capacities in order to accommodate the growing number of users. The demand for bandwidth-both wired and wireless-and innovative new bandwidth-intensive services is soaring. The use of high-definition video, real-time collaboration, e-commerce, social networking, and other multimedia Web applications is becoming increasingly important to individual users and businesses. The use of mobile broadband and file-sharing applications is rising sharply. Advances in research applications and the evolution to cloud networking are also causing bandwidth pressure on existing networks. This course will provide an in-depth understanding of the Internet architecture and underlying technologies and applications that address the challenges summarized above and provide services to users at high availability, reliability, and flexibility in a cost-effective manner. Specific topics to be discussed in this course include high-speed Internet requirements analysis, Internet architecture and protocols, convergence of mobile and terrestrial networks, high-speed LAN/WAN options and configurations, emerging and future switching and transmission technologies, and network virtualization. The course will also cover unique challenges to management and security of the high-speed Internets and how they are addressed. Other topics include emerging technologies and future trends.
Prerequisite(s): 605.671 Principles of Data Communications Networks.

EN.605.674. Network Programming. 3 Credits.
Emphasis is placed on the theory and practice associated with the implementation and use of the most common processto-process communications associated with UNIX. The interprocess communications comprise both local and distributed architectures. The distributed communications protocols include those most widely implemented and used: the worldwide Internet protocol suite [the Transmission Control Protocol/ Internet Protocol (TCP/IP), and the U.S. government-mandated International Organization for Standardization (ISO) protocol suite]. Practical skills are developed, including the ability to implement and configure protocol servers (daemons) and their clients. Students are expected to have working knowledge of UNIX.
Prerequisite(s): 605.671 Principles of Data Communications Networks, or 605.614 System Development in the UNIX Environment .

EN.605.675. Protocol Design. 3 Credits.
This course covers the formal design, specification, and validation of computer and network protocols. Design, implementation, and verification of protocols will be illustrated using the latest simulation tools, such as OPNET and NS2. Protocol examples include the latest wired and wireless networks, such as the IEEE 802.X family, as well as protocols in VoIP, Web 2.0, and network security. The course focuses on protocol specification, structured protocol design, protocol models, and protocol validation. Students will gain hands-on experience using simulation tools to design, validate, and assess protocols.
Prerequisite(s): 605.671 Principles of Data Communications Networks or equivalent.

EN.605.677. Internetworking with TCP/IP I. 3 Credits.
This course investigates the underlying technology of the Internet. The presentation begins with a survey of distributed applications operating over the Internet, including the Web, electronic mail, VoIP, instant messaging, file transfers and peerto-peer file sharing. The course investigates the details of the Internet architecture and the TCP/IP protocol suite, covering the protocols that provide communications services to end systems and the management and control protocols that create the Internet from disparate underlying networks and technologies. Communications-related protocols analyzed in detail include the foundational Internet Protocol (IP), the connection-oriented reliable Transmission Control Protocol (TCP), the connectionless User Datagram Protocol (UDP) and the Real-Time Protocol (RTP) for streaming media. To allow the student to understand the control and management of the Internet, the course analyzes protocols that support naming (DNS), addressing and configuration (DHCP), management (SNMP) and the dynamic IP routing protocols RIP, OSPF and BGP.
Prerequisite(s): 605.202 Data Structures; 605.671 Principles of Data Communications Networks.

EN.605.678. Next Generation Mobile Networks with 5G. 3 Credits.
The primary focus of this course is to introduce the next generation mobile networks, including both Cellular and WLAN technologies in great detail, to discuss various types of IP-based mobility protocols, namely Mobile-IP, Mobile IPv6, ProxyMIPv6, SIP-mobility, and Cellular IP, and to explore systems optimization techniques to support seamless handover during Inter RAT handover (e.g., 4G, 5G, and WLAN). Additionally, the course will briefly introduce the principles of cellular communications system and will then move on to describe the evolution of different generations of cellular systems including 2G, 3G, 4G, and 5G as being defined in 3GPP. At the same time it will discuss IEEE WLAN standards as developed by IEEE 802 working group including 802.11 (a, b, g, n) and 802.11 (ax, ay, ac). The Media Independent Handover standard IEEE 802.21 (e.g., integrating WLAN and 3G/4G cellular networks to provide session/service continuity) is also introduced. Further, the course will describe the 4G Long Term Evolution (LTE) in detail, covering its various components—namely Evolved UMTS Terrestrial Radio Access Network (E-UTRAN), EPC (Evolved Packet Network), and IMS (IP Multimedia Subsystem)—and all the associated interfaces and protocols, and the current efforts on 5G evolution and will touch upon various 5G pillars, namely SDN (Software Defined Networking), Network Function Virtualization, Cloud RAN, Network Slicing, Mobile Edge Cloud, and Edge Security. Finally, the course will highlight various standards activities within 3GPP, IEEE, IETF, NGMN, and ITU and will introduce some research problems for future study in the mobility area, presenting various deployment use cases and experimental results from the open-source testbeds.
Prerequisite(s): 605.202 Data Structures; 605.671 Principles of Data Communications Networks.

EN.605.681. Principles of Enterprise Web Development. 3 Credits.
This course examines three major topics in the development of applications for the World Wide Web. The first is website development using HTML and related standards. The second is the implementation of client-side applications using the Java programming language, including user interface development, asynchronously event handling, multithreaded programming, and network programming. Distributed object protocols via RMI or CorBA and distributed database access via JDBC may also be introduced. The third topic is the design of server-side web applications, for which students will examine the underlying web protocol (HTTP), the development of client-side interfaces (e.g., via HTML forms), and the implementation of server-side programs (e.g., via Java servlets or traditional CGI). Prerequisite(s): 605.202 Data Structures.

EN.605.682. Web Application Development with Java. 3 Credits.
This project-oriented course will enable students to use various techniques for building browser-based applications for dynamically generated websites, e-commerce, web-enabled enterprise computing, and other applications that require web access to server-based resources. Particular attention will be paid to methods for making web-based applications efficient, maintainable, and flexible. The course will use at least two sets of tools: servlets/JSP and a higher-level Java-based framework such as JSF 2.0. Major topics will include handling HTTP request information, generating HTTP response data, tracking sessions, designing custom tag libraries or components, page templating, asynchronously updating pages with Ajax, and separating content from presentation through use of the MVC architecture. Additional topics may include HTML5, database access techniques for web apps, web app security, and dependency injection in web apps (e.g., with the Spring framework). Course Note(s): Formerly 605.682 Web Application Development with Servlets and JavaServer Pages (JSP).
Prerequisite(s): 605.681 Principles of Enterprise Web Development or equivalent Java experience.

EN.605.683. Java Enterprise Development: Processes, Tools and Infrastructure. 3 Credits.
The focus of this course is to get the student acclimated to the process and tools used in the design to delivery cycle of an Enterprise Application using Java. It will introduce students to the use of build tools and repositories for creating and maintaining software in a team environment. It will then then cover tools and techniques for improving the quality of Enterprise Software like unit and integration testing, code optimization and profiling. It will also cover techniques for automation of processes in testing and deployment of software; like Continuous Integration and the use and orchestration with virtual containers. The course will also look at some modern integrated development environments and demonstrate how they integrate with the aspects of the class. A sample of tools covered in the class will include Maven, Gradle, JMeter, Postman, Jenkins, Git, JProfiler, Docker, Docker Compose, Eclipse and IntelliJ.
Prerequisite(s): EN.605.681 Principles of Enterprise Web Development with Java

EN.605.684. Agile Development with Ruby on Rails. 3 Credits.
Modern web applications are expected to facilitate collaboration, with user participation being a significant facet of the system. Components such as wikis, blogs, and forums are now commonplace. While feature sets continue to expand, there is continuing pressure to develop and deploy capabilities more quickly to enable organizations to remain competitive. This pressure has led to the development of languages and frameworks geared toward rapid prototyping, with Ruby on Rails being the most popular. Ruby on Rails is a model-view-controller (MVC) framework that enables efficient application development and deployment. Techniques such as convention over configuration and object-relational mapping with ActiveRecord along with enhanced AJAX support offer a simple environment with significant productivity gains. This code-intensive course introduces Ruby on Rails, the patterns it implements, and its applicability to the rapid development of collaborative applications.
Prerequisite(s): 605.681 Principles of Enterprise Web Development or equivalent; 605.202 Data Structures

EN.605.686. Mobile Application Development for the Android Platform. 3 Credits.
This project-oriented course will investigate application development for the Android mobile platform. We will look at techniques for building applications that adapt to the ways in which mobile apps differ from traditional desktop or web-based apps, including constrained resources, small screen sizes, varying display resolutions, intermittent network connectivity, specialized sensors, and security restrictions. We will explore best practices for making mobile applications flexible: using XML-based layouts, networking via NFC and Wi-Fi, determining device location and orientation, deploying applications, gracefully handling shutdowns and restarts to the application, embedding web components in applications, showing maps with the Google Maps plug-in, and storing local data with SQLite. Prerequisite(s): Expertise in simple SQL, Java and basic APIs, including callbacks, threads, XML, lists, and maps.. Course Note(s): Students should already be very comfortable with Java. Students will be provided links to download free tools for building and testing Android apps. Note that Android emulators may run quite slowly on some machines; physical Android devices are strongly recommended for this course.

EN.605.687. Mobile Application Development for the iOS Platform. 3 Credits.
This project-oriented course will investigate application development on iOS platforms. First, we will cover the main language for iOS, Swift, Apple's in-house, open sourced language for iOS and OS X development, along with tools such as Xcode, Interface Builder, Instruments, and Swift Playgrounds. Second, we will discuss the aspects of creating an application: the application life cycle, user experience and data presentation, user interface elements (including how to use the SwiftUI framework), and application performance. Then, we will discuss the application frameworks that the iOS SDK provides: CoreData, SpriteKit, MapKit, and Notifications, to name just a few. Finally, we will prepare your app for deployment, considering localization and internationalization, accessibility, and iTunes Connect. By the end of the class, students will be able to use Xcode, implement the Model-View-Controller paradigm, use Protocols and Delegates, construct a user interface that operates on many different devices, store and retrieve data on the network, interact with the OS or other applications, distinguish between the various iOS frameworks, and explain the App publication process.Course prerequisite(s): 605.201 Introduction to Programming Using Java or equivalent Java or Objective C experience.Course note(s): Access to a Mac running the current version of macOS is required for this class. Development tools can be downloaded for free from the Mac App Store. Additional hardware (iPhones, iPods, iPads) is strongly suggested, as several class examples and some projects will work best (or only work) on devices. *THIS REQUIREMENT IS SUBJECT TO CHANGE*

EN.605.701. Software Systems Engineering. 3 Credits.
Software Systems Engineering applies engineering principles and the system view to the software development process. The course focuses on the engineering of complex systems that have a strong software component. This course is based on the philosophy that the key to engineering a good software system lies just as much in the process that is followed as in the purely technical regime. The course will show how good a software development process is and how to make a software process better by studying successful techniques that have been employed to produce correct software systems within budget. Topics are explored in a sequence designed to reflect the way one would choose to implement process improvements. These topics include steps to initiate process change, methods to establish control over the software process, ways to specify the development process, methods for quantitative process control, and how to focus on problem prevention. Students will prepare term projects. Prerequisite(s): 605.202 Data Structures; one software engineering course beyond 605.601 Foundations of Software Engineering.

EN.605.702. Service-Oriented Architecture. 3 Credits.
Service-Oriented Architecture (SOA) is a way to organize and use distributed capabilities that may be controlled by different owners. SOA provides a uniform means to offer, discover, interact with, and use capabilities to produce desired effects consistent with specified preconditions and requirements. This course describes SOA concepts and design principles, interoperability standards, security considerations, runtime infrastructure and web services as an implementation technology for SOA. Given the focus on shared capabilities, SOA involves more than technology. Therefore, additional topics will include the impact of SOA on culture, organization, and governance. Prerequisite(s): 605.601 Foundations of Software Engineering and 605.704 Object-Oriented Analysis and Design or equivalent experience are highly recommended. **Prerequisite(s):** 605.601 Foundations of Software Engineering and 605.704 Object-Oriented Analysis and Design or equivalent experience are highly recommended.

EN.605.704. Object-Oriented Analysis and Design. 3 Credits.
This course describes fundamental principles of object-oriented modeling, requirements development, analysis, and design. Topics include specification of software requirements; objectoriented analysis approaches, including dynamic and static modeling with the Unified Modeling Language (UML v2); object-oriented design; object-oriented reuse, including design patterns; and software implementation concerns. Optional topics include the Systems Modeling Language (SysML), Object-Oriented Systems Engineering Methodology (OOSEM), managing object-oriented projects, and the Object Constraint Language (OCL). Prerequisite(s): Experience in object-oriented programming using a language such as Java or C++.

EN.605.705. Software Safety. 3 Credits.
This course describes how to develop and use software that is free of imperfections that could cause unsafe conditions in safety-critical systems. Systems engineering and software engineering techniques are described for developing "safeware," and case studies are presented regarding catastrophic situations that resulted from software and system faults that could have been avoided. Specific techniques of risk analysis, hazard analysis, fault tolerance, and safety tradeoffs within the software engineering paradigm are discussed. Prerequisite(s): 605.202 Data Structures.

EN.605.707. Software Patterns. 3 Credits.
Software patterns encapsulate the knowledge of experienced software professionals in a manner that allows developers to apply that knowledge to similar problems. Patterns for software are analogous to the books of solutions that enable electrical engineers and civil engineers to avoid having to derive every new circuit or bridge design from first principles. This course will introduce the concept of software patterns, and explore the wide variety of patterns that may be applied to the production, analysis, design, implementation, and maintenance of software. The format of the course will emphasize the discussion of patterns and their application. Each student will be expected to lead a discussion and to actively participate in others. Students will also be expected to introduce new patterns or pattern languages through research or developed from their own experience. Programming exercises performed outside of class will be used enhance discussion and illustrate the application of patterns.
Prerequisite(s): 605.604 Object-Oriented Programming with C++ or permission of instructor.

EN.605.708. Tools and Techniques of Software Project Management. 3 Credits.
This course examines tools and techniques used to lead software-intensive programs. Techniques for RFP analysis and proposal development are explored, and techniques of size estimation (function points, feature points, and lines-ofcode estimation) and the use of models such as COCOMO to convert size to effort and schedule are described. In addition, conversion of estimated effort to dollars and the effects of fringe, overhead, skill mix profiles, and staffing profiles on total dollar cost are explained. Moreover, techniques for estimating effort and planning the COTS intensive development programs are described, and tools and techniques for measuring process maturity and process efficiency (e.g., CMMi, Lean, Six Sigma, and Kaizen) are addressed. The course also investigates the formation and management of virtual teams, as well as techniques that can be used to ensure success in this environment. Finally, the course addresses topics that require collaboration between the project manager and human resources, such as personnel retention strategies, managing unsatisfactory performance, and formal mentoring programs. Prerequisite(s): Three to five years technical work experience is recommended.

EN.605.709. Seminar in Software Engineering. 3 Credits.
This course examines the underlying concepts and latest topics in software engineering. Potential topics include use of effective open-source software development techniques such as agile methods, automated code generation, testing strategies, development tools and environments, patterns, metrics in the development process, successful teamwork, and training aspects of CMMI. Each student will select and report on a software engineering topic, independently research a topic, and prepare a paper describing a major software engineering issue. The course is taught using a seminar format in which significant portions of the class period are set aside for students to lead and actively participate in discussions.
Prerequisite(s): One software engineering course beyond 605.601 Foundations of Software Engineering or permission of the instructor.

EN.605.715. Software Development for Real-Time Embedded Systems. 3 Credits.
This course examines the hardware and software technologies behind real-time, embedded computer systems. From smart kitchen appliances to sophisticated flight control for airliners, embedded computers play an important role in our everyday lives. Hardware topics include microcomputers and support devices (e.g., flash, ROM, DMA, timers, clocks, A/D, and D/A), as well as common applications (e.g., servo and stepper motor control, automotive sensors, and voice processing). Software topics focus on unique aspects of embedded programming and include interrupts, real-time control, communication, common design patterns, and special test considerations. The course also explores the unique tools that are used to develop and test embedded systems. Labs, beginning with using Bare Metal and Free RTOS on Arduino for simple devices and culminating with using Linux on Raspberry-Pi for Quad-Copter flight control, are developed.

EN.605.716. Modeling and Simulation of Complex Systems. 3 Credits.
This multi-disciplinary course focuses on the application of modeling and simulation principles to complex systems. A complex system is a large-scale nonlinear system consisting of interconnected or interwoven parts (such as a biological organism, an ecological system, the economy, fluids or strongly-coupled solids). The subject is interdisciplinary with foundations in mathematics, nonlinear science, numerical simulations and statistical physics. The course begins with an overview of complex systems, followed by modeling techniques based on nonlinear differential equations, networks, and stochastic models. Simulations are conducted via numerical calculus, analog circuits, Monte Carlo methods, and cellular automata. In the course we will model, program, and analyze a wide variety of complex systems, including dynamical and chaotic systems, cellular automata, and iterated functions. By defining and iterating an individual course project throughout the term, students will gain hands-on experience and understanding of complex systems that arise from combinations of elementary rules. Students will be able to define, solve, and plot systems of linear and non-linear systems of differential equations and model various complex systems important in applications of population biology, epidemiology, circuit theory, fluid mechanics, and statistical physics.Course prerequisite(s): Knowledge of elementary probability and statistics and previous exposure to differential equations. Students applying this course to the MS in Bioinformatics should also have completed at least one Bioinformatics course prior to enrollment.Course note(s): This course may be counted toward a three-course concentration in Bioinformatics.

EN.605.721. Design and Analysis of Algorithms. 3 Credits.
In this follow-on course to 605.621 Foundations of Algorithms, design paradigms are explored in greater depth, and more advanced techniques for solving computational problems are presented. Topics include randomized algorithms, adaptive algorithms (genetic, neural networks, simulated annealing), approximate algorithms, advanced data structures, online algorithms, computational complexity classes and intractability, formal proofs of correctness, sorting networks, and parallel algorithms. Students will read research papers in the field of algorithms and will investigate the practicality and implementation issues with state-of-the-art solutions to algorithmic problems. Grading is based on problem sets, programming projects, and in-class presentations.
Prerequisite(s): 605.621 Foundations of Algorithms or equivalent; 605.203 Discrete Mathematics or equivalent.

EN.605.724. Applied Game Theory. 3 Credits.
In many organizations in the private and the public sectors, there is a need to support complex decisions that include a game-theoretic aspect. These decisions impact activities ranging from tactical to strategic, and play out in a number of problems, including monitoring and management of ongoing operations, the dynamics of organizational relationships in the competitive environment, and military force planning. This course extends treatment of game theoretic concepts and constructs, and explores their implementation and application, highlighting key issues such as decision space exploration and analysis, visualization, and the creation and use of models for specific domains. Students will have the opportunity to design a course project based on their area of professional or personal interest.

EN.605.725. Queuing Theory with Applications to Computer Science. 3 Credits.
Queues are a ubiquitous part of everyday life; common examples are supermarket checkout stations, help desks call centers, manufacturing assembly lines, wireless communication networks, and multitasking computers. Queuing theory provides a rich and useful set of mathematical models for the analysis and design of service process for which there is contention for shared resources. This course explores both theory and application of fundamental and advanced models in this field. Fundamental models include single- and multipleserver Markov queues, bulk arrival and bulk service processes, and priority queues. Applications emphasize communication networks and computer operations, but may include examples from transportation, manufacturing, and the service industry. Advanced topics may vary. Prerequisite(s): Multivariate calculus and a graduate course in probability and statistics such as 625.603 Statistical Methods and Data Analysis or equivalent. Course Note(s): This course is the same as 625.734 Queuing Theory with Applications to Computer Science.

EN.605.726. Game Theory. 3 Credits.
Game theory is a field of applied mathematics that describes and analyzes interactive decision making when two or more parties are involved. Since finding a firm mathematical footing in 1928, it has been applied to many fields, including economics, political science, foreign policy, and engineering. This course will serve both as an introduction to and a survey of applications of game theory. Therefore, after covering the mathematical foundational work with some measure of mathematical rigor, we will examine many real-world situations, both historical and current. Topics include two-person/N-person game, cooperative/non-cooperative game, static/dynamic game, combinatorial/strategic/coalitional game, and their respective examples and applications. Further attention will be given to the meaning and the computational complexity of finding of Nash equilibrium.Prerequisite(s): Multivariate calculus, linear algebra and matrix theory (e.g., 625.609 Matrix Theory), and a course in probability and statistics (such as 625.603 Statistical Methods and Data Analysis). Course Note(s): This course is the same as 625.741 Game Theory.

EN.605.727. Computational Geometry. 3 Credits.
This course covers fundamental algorithms for efficiently solving geometric problems, especially ones involving 2D polygons and 3D polyhedrons. Topics include elementary geometric operations; polygon visibility, triangulation, and partitioning; computing convex hulls; proximity searching, Voronoi diagrams, and Delaunay triangulations with applications; special polygon and polyhedron algorithms such as point containment and extreme point determination; point location in a planar graph subdivision; dimension reduction in data; and robot motion planning around polygon obstacles. Applications to such areas as computer graphics, big data analytics and pattern recognition, geometric databases, numerical taxonomy, and robotics will be addressed. The course covers theory to the extent that it aids in understanding how the algorithms work. Emphasis is placed on algorithm design and implementation. Programming projects are an important part of the coursework. Prerequisite(s): Foundations of algorithms. Some familiarity with linear algebra.
Prerequisite(s): 605.621 Foundations of Algorithms. Some familiarity with linear algebra.

EN.605.728. Quantum Computation. 3 Credits.
Scalable quantum computers aren't here yet. But recent progress suggests they may be on their way, and that it is now time to start planning for their potential impact: NSA announced in 2015 a shift in focus from elliptic curve to quantum resistant cryptography, and NIST has initiated a large-scale study of postquantum cryptography. This course provides an introduction to quantum computation for computer scientists: the focus is on algorithms rather than physical devices, and familiarity with quantum mechanics (or any physics at all) is not a prerequisite. Instead, pertinent aspects of the quantum mechanics formalism are developed as needed in class. The course begins with an introduction to the QM formalism. It then develops the abstract model of a quantum computer, and discusses how quantum computers enable us to achieve, for some problems, a significant speedup (in some cases an exponential speedup) over any known classical algorithm. This discussion provides the basis for a detailed examination of quantum integer factoring, quantum search, and other quantum algorithms. The course also explores quantum error correction, quantum teleportation, and quantum cryptography. It concludes with a glimpse at what the cryptographic landscape will look like in a world with quantum computers. Required work includes problem sets and a research project. Prerequisite(s): Some familiarity with linear algebra and with the design and analysis of algorithms.

EN.605.729. Formal Methods. 3 Credits.
Formal verification of a program is the mathematical proof that it does what is expected of it. The 21st century has seen a vast worldwide interest in formal methods. Four journals (Automated Reasoning, Logic and Algebraic Programming, Formalized Mathematics, and Science of Computer Programming) and over a dozen yearly conferences, each of which has been held at least since 2000, are specifically devoted to these matters. Centers of ongoing formal methods research include Argonne, Berkeley, Bialystok (Poland), Cambridge, Clemson, HP, INRIA, Iowa State, Karlsruhe, Lausanne, Microsoft, MITRE, Munich, NYU, Penn, Praxis, and SRI. Methods have been developed for Java (JML), Ada (SPARC), C#, C, and Eiffel (Spec#), Haskell, Ocaml, and Scheme (Coq), Pascal (Sunrise), Modula-3 (ESC), and a number of special-purpose languages. This course is an introduction to this vast world of formal methods. Our concern will be the formal verification of the widest possible variety of programming language features and techniques. Each student will carry out an investigation of one or another of the existing formal verification systems, applying it to a program of the student's choice.

EN.605.731. Survey of Cloud Computing Security. 3 Credits.
The promise of significant cost savings and inherent flexibility of resources are an impetus for the adoption of cloud computing by many organizations. Cloud computing also introduces privacy and security risks that are not traditionally present in a siloed data center. This course focuses on these security concerns and countermeasures for a cloud environment. An overview of cloud computing and virtualization, the critical technology underpinning cloud computing, provides the necessary background for these threats. Additional topics vary but may include access control, identity management, denial of service, account and service hijacking, secure APIs, malware, forensics, regulatory compliance, trustworthy computing, and secure computing in the cloud. This course follows a seminar-style format where students are expected to lead class discussions and write a publication-quality paper as part of a course project.

EN.605.741. Large-Scale Database Systems. 3 Credits.
This course investigates the theory and practice of modern large-scale database systems. Large-scale approaches include distributed relational databases; data warehouses; and non-relational databases including HDFS, Hadoop, Accumulo for query and graph algorithms, and Mahout bound to Spark for machine learning algorithms. Topics discussed include data design and architecture; database security, integrity, query processing, query optimization, transaction management, concurrency control, and fault tolerance; and query formulation, graph algorithms, and machine learning algorithms on large-scale distributed data systems. At the end of the course, students will understand the principles of several common large-scale data systems including their architectures, performance, and costs. Students will also gain a sense of which approach is recommended for different requirements and circumstances. **Prerequisite(s):** 605.202 Data Structures; 605.641 Principles of Database Systems or equivalent. Familiarity with "big-O" concepts and notation is recommended.

EN.605.742. Deep Neural Networks.
Prerequisite(s): EN.605.401Foundations of Software Engineering AND EN.605.411Foundations of Computer Architecture AND EN.605.421Foundations of Algorithms ANDEN.605.481Distributed Development on the WWW

EN.605.744. Information Retrieval. 3 Credits.
A multibillion-dollar industry has grown to address the problem of finding information. Commercial search engines are based on information retrieval: the efficient storage, organization, and retrieval of text. This course covers both the theory and practice of text retrieval technology. Topics include automatic index construction, formal models of retrieval, Internet search, text classification, multilingual retrieval, question answering, and related topics in NLP and computational linguistics. A practical approach is emphasized and students will complete several programming projects to implement components of a retrieval engine. Students will also give a class presentation based on an independent project or a research topic from the IR literature. Prerequisite(s): 605.202 Data Structures.

EN.605.745. Reasoning Under Uncertainty. 3 Credits.
This course is concerned with the problems of inference and decision making under uncertainty. It develops the theoretical basis for a number of different approaches and explores sample applications. The course discusses foundational issues in probability and statistics, including the meaning of probability statement, and the necessity of a rational agent acting in accord with probability theory. We will look at possible generalizations of Bayesian probability, including Dempster-Shafer theory. Next, we will develop algorithms for Bayesian networks—graphical probabilistic models—for exact and approximate inference and consider several application areas. Finally, the course will examine the problem of making optimal decisions under uncertainty. We will explore the conceptual foundations of decision theory and then consider influence diagrams, which are graphical models extending Bayesian networks to the domain of decision analysis. As time permits, we will also look at Bayesian games and Markov decision processes. Pertinent background in probability and theoretical computer science is developed as needed in the course.

EN.605.746. Advanced Machine Learning. 3 Credits.
This course focuses on recent advances in machine learning and on developing skills for performing research to advance the state of knowledge in machine learning. The material integrates multiple ideas from basic machine learning and assumes familiarity with concepts such as inductive bias, the bias-variance trade-off, the curse of dimensionality, and no free lunch. Topics range from determining appropriate data representations and models for learning, understanding different algorithms for knowledge and model discovery, and using sound theoretical and experimental techniques in assessing learning performance. Specific approaches discussed cover nonparametric and parametric learning; supervised, unsupervised, and semisupervised learning; graphical models; ensemble methods; and reinforcement learning. Topics will be discussed in the context of research reported in the literature within the previous three years. Students will participate in seminar discussions and will present the results of their individual research project. Prerequisite(s): 605.649 Introduction to Machine Learning; multivariate calculus, linear algebra, probability and statistics, discrete mathematics.
Prerequisite(s): 605.649 Introduction to Machine Learning; multivariate calculus, linear algebra, probability and statistics, discrete mathematics.

EN.605.747. Evolutionary Computation. 3 Credits.
Recently, principles from the biological sciences have motivated the study of alternative computational models and approaches to problem solving. This course explores how principles from theories of evolution and natural selection can be used to construct machines that exhibit nontrivial behavior. In particular, the course covers techniques from genetic algorithms, genetic programming, and artificial life for developing software agents capable of solving problems as individuals and as members of a larger community of agents. Specific topics addressed include representation and schemata; selection, reproduction, and recombination; theoretical models of evolutionary computation; optimal allocation of trials (i.e., bandit problems); search, optimization, and machine learning; evolution of programs; population dynamics; and emergent behavior. Students will participate in seminar discussions and will complete and present the results of an individual project.

EN.605.748. Semantic Natural Language Processing. 3 Credits.
This course introduces the fundamental concepts underlying knowledge representation, semantics, and pragmatics in natural language processing. Students will gain an in-depth understanding of the techniques central to computational semantics and discourse for processing linguistic information. The course examines semantic NLP models and algorithms using both the traditional symbolic and the more recent statistical approaches. The course also covers the development of modern NLP systems capable of carrying out dialogue and conversation.
Prerequisite(s): 605.645 Artificial Intelligence or equivalent Course Note(s): This course and 605.646 Natural Language Processing can be taken independently of each other.

EN.605.751. Algorithms for Structural Bioinformatics. 3 Credits.
This course is an interdisciplinary approach to the concepts, principals, computational methods and algorithms used in structural bioinformatics. It focuses on the fundamental aspects of structural biology along with computational methods and algorithms for studying protein folding, structure prediction and analysis. Algorithms for the prediction and annotation of protein secondary and tertiary structure and for structurestructure comparison will be studied in depth. We will also show how such algorithms and methods can be adapted for use with nucleic acids structure prediction and analysis. Students will apply various software tools and structure-visualization software to protein structure prediction and structurestructure comparison. Prerequisite(s): 605.205 Molecular Biology for Computer Scientists or equivalent. 605.661 Principles of Bioinformatics is recommended.
Prerequisite(s): 605.205 Molecular Biology for Computer Scientists or equivalent. 605.661 Principles of Bioinformatics is recommended.

EN.605.754. Analysis of Gene Expression and High-Content Biological Data. 3 Credits.
The development of microarray technology, rapid sequencing, protein chips, and metabolic data has led to an explosion in the collection of "high-content" biological data. This course explores the analysis and mining of gene expression data and high-content biological data. A survey of gene and protein arrays, laboratory information management systems, data normalization, and available tools is followed by a more in-depth treatment of differential gene expression detection, clustering techniques, pathway extraction, network model building, biomarker evaluation, and model identification. Both clinical and research data will be considered. The student will develop skills in statistical analysis and data mining including statistical detection theory, nonlinear and multiple regression, entropy measurement, detection of hidden patterns in data, heuristic search and learning algorithms. Applied mathematical concepts and biological principles will be introduced, and students will focus on algorithm design and software application for designing and implementing novel ways of analyzing gene, protein and metabolic expression data. The statistical programming language R is used extensively in lecture and homework. Packages from Bioconductor, including many which contain data sets, are used regularly as well. Students will complete data analysis assignments individually and in small teams. Prerequisite(s): 605.205 Molecular Biology for Computer Scientists or equivalent or a prior course in Bioinformatics, a course in probability and statistics, and ability to program in a high-level language. Course Note(s): There are no exams, but programming assignments are intensive. Students in the MS Bioinformatics program may take both this course and 410.671 Microarrays and Analysis, as the content is largely mutually exclusive
Prerequisite(s): 605.205 Molecular Biology for Computer Scientists or equivalent or a prior course in Bioinformatics, a course in probability and statistics, and ability to program in a high-level language.

EN.605.755. Systems Biology. 3 Credits.
Systems biology is the study of complex biological systems using theoretical, mathematical, and computational tools and concepts. The advent of genomics, big data, and highpowered computing is allowing better understanding and elucidation of these systems. Central to systems biology is the development of computational models, based on sound statistics, which incorporate biological structures and networks, and can be informed and tested, with data on multiple scales. In this class, students will learn to develop and use different types of models of complex biological systems and how to test and perturb them. Students will learn basic biological system components and dynamics, as well as the data formats, sources, and modeling tools required to interrogate them. Tools will be used relating to functional genomics, evolution, biochemical systems, and cell biology. Students will utilize a model they have developed and available data from public repositories to investigate both a discovery-based project and a hypothesisbased project. Prerequisite(s): Courses in molecular biology (605.205 Molecular Biology for Computer Scientists or 410.602 Molecular Biology) and differential equations.
Prerequisite(s): Courses in molecular biology (605.205 Molecular Biology for Computer Scientists or 410.602 Molecular Biology) and differential equations.

EN.605.759. Independent Project in Bioinformatics. 3 Credits.
This course is for students who would like to carry out a significant project in bioinformatics as part of their graduate program. The course may be used to conduct minor research, an in-depth literature survey, or a software implementation related to recent developments in the field. Students who enroll in this course are encouraged to attend at least one industry conference in bioinformatics related to their area of study. To enroll in this course, the student must be within two courses of degree completion and must obtain the approval and support of a sponsoring faculty member. Course Note(s): A student may not receive credit for both 605.759 and 605.802 Independent Study in Computer Science II.

EN.605.767. Applied Computer Graphics. 3 Credits.
This course examines advanced rendering topics in computer graphics. The course focuses on the mathematics and theory behind 3D graphics rendering. Topics include 3D surface representations including fractal geometry methods; visible surface detection and hidden surface removal; and surface rendering methods with discussion of lighting models, color theory, texturing, and ray tracing. Laboratory exercises provide practical application of these concepts. The course also includes a survey of graphics rendering applications (animation, modeling and simulation, and realistic rendering) and software. Students perform laboratory exercises using the C++ programming language.
Prerequisite(s): 605.667 Computer Graphics or familiarity with three-dimensional viewing and modeling transformations.

EN.605.771. Wired and Wireless Local and Metropolitan Area Networks. 3 Credits.
This course provides a detailed examination of wired and wireless local and metropolitan area network (LAN and MAN) technologies, protocols, and the methods used for implementing LAN- and MAN-based enterprise intranets. The structure and operation of the IEEE 802 media access control (MAC) and physical layer protocols are examined in detail. The 802.2 logical link control, 802.3/Ethernet, 802.4 token bus, and 802.5 token ring protocols are analyzed, and the construction of LAN-based enterprise intranets is examined through a detailed analysis of bridging, routing, and switching techniques. High-speed LAN technologies are discussed through an examination of FDDI, Fast Ethernet, 100VG AnyLAN, ATM LAN Emulation (LANE), and Fibre Channel protocols, along with the new standards for gigabit and 10-gigabit Ethernet. In addition, the 802.6 DQDB and 802.17 Resilient Packet Ring MAN protocols are discussed. Finally, the new and emerging wireless LAN and MAN standards are examined. The 802.11 (Wi-Fi) wireless LAN and 802.15 (Bluetooth) wireless PAN standards are discussed in detail along with the emerging 802.16 (WiMAX) wireless MAN standard. Topics include Manchester and Differential Manchester encoding techniques; bus, star, and ring topologies; optical fiber, coaxial cable, and UTP media; baseband, broadband, and carrierband bus networks; hubs, switched LANs, and full duplex LANs; VLANs and prioritization techniques; transparent and source routing bridge algorithms; packet bursting and carrier extension schemes; wireless spread spectrum and frequency hopping transmission techniques; wireless collision avoidance media access control; and security schemes. Students may use the network lab to configure LAN switches and Cisco routers, as well as to observe the interconnection of LAN networks.
Prerequisite(s): 605.202 Data Structures; 605.671 Principles of Data Communications Networks or 635.611 Principles of Network Engineering.

EN.605.772. Network Security Management. 3 Credits.
Information transfer speeds and infrastructure capacities must continue to evolve to support not only traditional voice and data but also multimedia services such as high-definition video, real-time collaboration, e-commerce, and social networking. While services are provided across terrestrial and mobile networks transparently to users, new technologies such as cloud computing efficiently make the services available to users irrespective of their geographic locations. In this rapidly evolving technological environment, network and security management (NSM) is the key to providing network access and connectivity, ensuring high availability of applications and services, and assuring users of the reliability and security of their transported information. Network Management (NM) encompasses all the activities, methods, operational procedures, tools, communications interfaces, protocols, and human resources pertaining to the operation, administration, maintenance, provisioning, and growth planning of communications networks. Security Management (SM) pertains to monitoring and control of security services and mechanisms including identification, authentication, authorization, access control, confidentiality, intrusion detection, correction, and prevention in order to protect the communications network infrastructure and services. NSM includes setting, monitoring, and maintaining certain performance metrics to ensure high performance levels and quality of service (QoS) to the users, along with support for infrastructure architecture and security planning, design, and implementation.
This course examines NSM standards, technologies, tools, industry best practices, and case studies, NSM areas that can be automated through expert systems, current issues, and future trends to adapt to emerging and evolving Internet technologies. Specific Internet and telecommunications standards discussed in depth in this course include SNMPv1, SNMPv2, SNMPv3, RMON, and OSI. Students will apply the standards, architectures, tools, and techniques learned in the course, as well as research state-of-the-art technologies in a team project.
Prerequisite(s): 605.771 Wired and Wireless Local and Metropolitan Area Networks, or 605.672 Computer Network Architectures and Protocols, or 605.677 Internetworking with TCP/IP I, or 635.611 Principles of Network Engineering.

EN.605.775. Optical Networking Technology. 3 Credits.
The Internet has hundreds of millions of users, is growing rapidly, and continues to evolve to accommodate an increasing number of voice, data, video, and imagery applications with diverse service requirements. Internet Protocol (IP) is the primary unifying protocol converging these applications and services over the Internet. The Internet's evolution has been accompanied by exponentially growing traffic volume on the network infrastructure. Optical networks are ideally suited to carry such large volumes of traffic, and the next generation of optical networks will be optimized for delivery of IP services while providing capacity in the range of terabits per second in a scalable and flexible way to support services such as voice over IP (VoIP) and IP television (IPTV). This course provides an in-depth understanding of existing and emerging optical network technologies. Specific topics covered include basics of fiber optic communications, SONET, DWDM, optical Ethernet, FTTB, FTTH, optical wavelength switching, IP over optical networks, MPLS, and GMPLS. Additional topics that may be discussed include optical network standards, network control and management, static and dynamic service provisioning, optical network design, and future directions.
Prerequisite(s): 605.673 High-Speed Internet Architecture, Technologies, and Applications or permission of the instructor.

EN.605.776. Fourth Generation Wireless Communications: WiMAX and LTE. 3 Credits.
This course compares the WiMAX and LTE fourth-generation (4G) technologies and their performance. An overview of the IEEE 802.16 standards (802.16d/e/j/m/n/p) and WiMAX Forum (Fixed WiMAX vs. Mobile WiMAX, Interoperability certification and Core network) is presented along with the 3GPP standards for LTE and LTE-Advanced as well as LTE network architecture. For WiMAX, the MAC, call flow, 2D resource map, QoS, and scheduling are presented. For LTE, both control plane and data plane protocols for Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and Evolved Packet Core (EPC) are presented. The topics include protocol architecture, bearer management, signaling, radio resource control (RRC), packet data convergence protocol (PDCP), radio link control (RLC), and MAC. In addition, the role of universal subscriber identity module (USIM), eNodeB, mobility management entity (MME), serving gateway (S-GW), packet data network gateway (P-GW), and home subscription server (HSS) as well as the call flow across these various nodes will be presented. The 2D resource grid along with QoS and scheduling will be explained in detail. The voice over LTE (VoLTE), self-organizing network (SON), LTE-direct, and LTE-Advanced [including coordinated multipoint (CoMP), carrier aggregation, and Intercell interference coordination (ICIC)] will be presented. Finally, spectrum considerations as well as the concept of white space and dynamic spectrum access (DSA) will be discussed. LTE security will be discussed in detail. The course will also highlight some of the Open Source LTE projects, and will discuss the experimental results from various testbeds.
Prerequisite(s): 605.202 Data Structures; 605.671 Principles of Data Communications Networks or 635.611 Principles of Network Engineering and another course in the Data Communications and Networking track.

EN.605.777. Internetworking with TCP/IP II. 3 Credits.
This course builds on the foundation established in 605.677, Internetworking with TCP/ IP I. Changes are being made in the infrastructure, operation, and protocols of the Internet to provide the performance and services needed for real-time applications. This course first examines the current architecture and operation of the Internet. The classful addressing concept will be introduced and the mapping of Internet addresses to physical addresses is discussed along with the extensions that have been made to the addressing paradigm, including subnet addressing, classless addressing, and network address translation. The performance enhancements being developed to provide quality of service (QoS) over the Internet and to provide faster routing through the use of IP switching techniques are discussed. Techniques for providing multicasting and mobility over the Internet are examined. Security considerations are addressed by examining Virtual Private Networks and the use of IP Security (IPSec) protocols. The next generation IP protocol (IPv6) is introduced, and the changes and enhancements to the IP protocol operation and to the addressing architecture are discussed in detail. Finally, the development of the Voice Over IP (VoIP) application and the convergence of circuit switching and packet switching are discussed. Topics include subnet addressing, CIDR, DHCP, DNS, NAT, IntServ, DiffServ, RSVP, CIP, MPOA, IP Switching, Tag Switching, MPLS, IP Multicast, IGMP, Reliable Musticast, Multicast Routing Protocols, IP Mobility Home Agents and Foreign Agents, Message Tunneling, Proxy and Gratuitous ARP, VPN Tunneling, PPTP, L2F, L2TP and SOCKSv5, VPN security, IPSec, Encapsulating Security Payload header, Authentication Header, Security Association, IPv6 Addressing, IPv6 protocol and extension headers, Neighbor Discovery, IPv6 Stateless Address Autoconfiguration, DHCPv6, VoIP, H.323 Gateways and Gatekeeper, SIP, SDP, RTP, MGCP, Megaco/H.248.
Prerequisite(s): 605.202 Data Structures; 605.677 Internetworking with TCP/IP I.

EN.605.778. Voice Over IP. 3 Credits.
The Internet has been growing exponentially and continues to evolve to accommodate an increasingly large number of applications with diverse service requirements. A remarkable aspect of this evolution is the convergence of real-time communications services with traditional data communications services over the Internet. In particular, Internet Telephony, or Voice Over IP is one of the most promising services currently being deployed. While there are many benefits to Voice over IP such as cost effectiveness and enhanced features, there exist a number of barriers to the widespread deployment of Internet Telephony. The purpose of this course is to provide in-depth understanding of the concept and operation of Voice Over IP and discuss issues and strategies to address the issues. In this course, students will gain understanding of how to adapt an IP packet network, which is basically designed for data, to provide wide-area voice communications. Topics include telephony fundamentals, Voice Over IP concepts, adapting IP networks to support voice, H.323 and SIP signaling protocols, QoS issues in IP networks, IETF standards, and network management. Prerequisite(s): 605.202 Data Structures; 605.677 Internetworking with TCP/IP I or 605.673 High-Speed Internet Architecture, Technologies, and Applications, or significant Internet technology-related work experience.
Prerequisite(s): 605.202 Data Structures; 605.677 Internetworking with TCP/IP I or 605.673 High-Speed Internet Architecture, Technologies, and Applications, or significant Internet technology-related work experience.

EN.605.779. Network Design and Performance Analysis. 3 Credits.
Networking services are a staple of our daily life. Different types of networks surround us all day long. This ubiquitous networking, thanks to smartphones and tablet computers, gives us the convenience of information at our fingertips. The right network architecture provides the fundamental support for network services, such as the products from Facebook, Google, Apple, etc. This course covers the details of network design and the design process. Starting from requirement specifications, a detail flow analysis is introduced. Examples of different network architecture designs, both in wireline and wireless, will be discussed, including mobile Ad Hoc network (MANET), mesh network, 4G cellular networks, wide area network (WAN), cloud networks, and advanced software define networking (SDN). Performance analyses and network security aspects are considered at every step of the design. Secured architecture covers Virtual Private Network (VPN) and Transport Layer Security (TLS)-based systems, with details on firewall and intrusion detection configurations. The course encourages hands-on projects selected from real network system problems.

EN.605.784. Enterprise Computing with Java. 3 Credits.
This comprehensive course explores core application aspects for developing, configuring, securing, deploying, and testing a Java-based service using a layered set of modern frameworks and libraries that can be used to develop full services and microservices to be deployed within a container. The emphasis of this course is on the center of the application (e.g., Spring, Spring Boot, Spring Data, and Spring Security) and will lay the foundation for other aspects (e.g., API, SQL and NoSQL data tiers, distributed services) covered in related courses.Students will learn thru lecture, examples, and hands-on experience in building multi-tier enterprise services using a configurable set of server-side technologies.Students will learn to:* Implement flexibly configured components and integrate them into different applications using inversion of control, injection, and numerous configuration and auto-configuration techniques* Implement unit and integration tests to demonstrate and verify the capabilities of their applications using JUnit and Mockito* Implement basic API access to service logic using modern REST-based approaches that include JSON and XML* Implement basic data access tiers to relational and NoSQL databases using the Spring Data framework* Implement security mechanisms to control access to deployed applications using the Spring Security framework
Prerequisite(s): 605.202 Data Structures; 605.681 Principles of Enterprise Web Development or equivalent. Course Note(s): Students will be assumed to already have strong Java skills and to be comfortable with IDEs.

EN.605.785. Web Services with SOAP and REST: Frameworks, Processes, and Applications. 3 Credits.
Web services is a technology, process, and software paradigm to extend the web from an infrastructure that provides services for humans to one that supports business integration over the web. This course presents concepts, features, and architectural models of web services from three perspectives: framework, process, and applications. Students will study three emerging standard protocols: Simple Object Access Protocol (SOAP); Web Services Description Language (WSDL); and Universal Description, Discovery, and Integration (UDDI). In contrast, Representational State Transfer (REST) is an architectural style for designing networked applications and exposing web services. REST delivers simplicity and true interoperability and is an alternative to complex mechanism such as CORBA, RPC, or SOAP-based web services and allows using simple HTTP to make calls between machines. The course will explain the REST principles and show how to use the Java standards for developing applications using RESTful API. Students will learn the benefits of and the technical architecture for using REST in applications, including how to design, build, and test RESTful services using Java and JAX-RS. This includes the role of key technologies such as HTTP, Extensible Markup Language (XML), and JavaScript Object Notation (JSON). Students also learn how to consume RESTful services in applications, including the role of JavaScript and Ajax, and how the RESTful approach differs from the SOAP-based approach, while comparing and contrasting the two techniques. Finally, the course will review other web services specifications and standards, and it will describe the use of web services to resolve business applications integration issues. WS-I Basic Profile and other guidance documents and recommended practices will be discussed in the context of achieving high levels of web services interoperability.
Prerequisite(s): 605.202 Data Structures; 605.644 XML Design Paradigms or equivalent XML and Java programming experience; knowledge of the J2EE platform and programming model is recommended.

EN.605.786. Enterprise System Design and Implementation. 3 Credits.
This course explores enterprise architectures for the development of scalable distributed systems. Effective patterns for distributed data access, MVC-based web tiers, and business logic components are explored as students build complex applications. Factors such as caching and clustering that enable distributed systems to scale to handle potentially thousands of users are a primary focus. In addition, creating a reusable blueprint for an enterprise architecture will be discussed. Applications developed utilizing these concepts are selected from current research topics in information retrieval, data visualization, and machine learning.
Prerequisite(s): 605.202 Data Structures; 605.784 Enterprise Computing with Java, 605.707 Software Patterns, or equivalent experience is recommended.

EN.605.787. Front End Web App Development. 3 Credits.
Using a web browser to access online resources is convenient because it provides universal access from any computer on any operating system in any location. Unfortunately, it often results in a poor user experience because HTML is a weak and noninteractive display language and HTTP is a weak and inefficient protocol. Full-fledged browser-embedded programs (e.g., ActiveX components, Java applets) have not succeeded in penetrating the market adequately, so a new class of applications has grown up that uses only the capabilities already available in most browsers. These applications were first popularized by Google but have since exploded in popularity throughout the developer community. The techniques to implement them were based on a group of technologies collectively known as Ajax, and the resultant applications were richer than the relatively static pure-HTML-based web applications that preceded them. These applications have become known as Ajax applications, rich internet applications, or Web 2.0 applications. This course will examine techniques to develop and deploy Ajax applications. We will look at the underlying techniques, then explore client-side tools (e.g., jQuery), server-side tools (e.g., JSON-RPC), and hybrid tools (e.g., the Google Web Toolkit) to simplify the development process. As we delve into several popular client and server-side libraries, we will be examining and paying attention to issues of usability, efficiency, security, and portability.
Prerequisite(s): Prerequisite(s): 605.202 Data Structures; 605.682 Web Application Development with Java or equivalent servlet and JSP experience.

EN.605.788. Big Data Processing Using Hadoop. 3 Credits.
Organizations today are generating massive amounts of data that are too large and too unwieldy to fit in relational databases. Therefore, organizations and enterprises are turning to massively parallel computing solutions such as Hadoop for help. The Apache Hadoop platform, with Hadoop Distributed File System (HDFS) and MapReduce (M/R) framework at its core, allows for distributed processing of large data sets across clusters of computers using the map and reduce programming model. It is designed to scale up from a single server to thousands of machines, offering local computation and storage. The Hadoop ecosystem is sizable in nature and includes many subprojects such as Hive and Pig for big data analytics, HBase for real-time access to big data, Zookeeper for distributed transaction process management, and Oozie for workflow. This course breaks down the walls of complexity of distributed processing of big data by providing a practical approach to developing applications on top of the Hadoop platform. By completing this course, students will gain an in-depth understanding of how MapReduce and Distributed File Systems work. In addition, they will be able to author Hadoop-based MapReduce applications in Java and also leverage Hadoop subprojects to build powerful data processing applications. Course Note(s): This course may be counted toward a threecourse track in Data Science and Cloud Computing.
Prerequisite(s): 605.202 Data Structures; 605.681 Principles of Enterprise Web Development or equivalent Java experience.

EN.605.795. Capstone Project in Computer Science. 3 Credits.
This course permits graduate students in computer science to work with other students and a faculty mentor to explore a topic in depth and apply principles and skills learned in the formal computer science courses to a real world problem. Students will work in self-organized groups of two to five students on a topic selected from a published list. Since students will have selected different courses to meet degree requirements, students should consider the combined strengths of the group in constituting their team. Each team will prepare a proposal, interim reports, a final report, and an oral presentation. The goal is to produce a publication quality paper and substantial software tool. This course has no formal content; each team should meet with their faculty mentor at least once a week and is responsible for developing their own timeline and working to complete it within one semester. The total time required for this course is comparable to the combined class and study time for a formal course.Course prerequisite(s): Seven computer science graduate courses including two courses numbered 605.7xx, all CS foundation courses, and meeting the track requirement; or admission to the post-master's certificate program. Students must also have permission of a faculty mentor, the student's academic advisor, and the program chair.Course note(s): Students may not receive graduate credit for both 605.795 and 605.802 Independent Study in Computer Science II. This course is only offered in the spring.

EN.605.801. Independent Study in Computer Science I. 3 Credits.
This course permits graduate students in computer science to work with a faculty mentor to explore a topic in depth or conduct research in selected areas. Requirements for completion include submission of a significant paper or project. Prerequisite(s): Seven computer science graduate courses including the foundation courses, three track-focused courses, and two courses numbered 605.7xx, or admission to the post-master's certificate. Students must also have permission of a faculty mentor, the student's academic advisor, and the program chair.

EN.605.802. Independent Study in Computer Science II. 3 Credits.
Students wishing to take a second independent study in computer science should sign up for this course. Prerequisite(s): 605.801 Independent Study in Computer Science I and permission of a faculty mentor, the student's academic advisor, and the program chair. Course Note(s): A student may not receive credit for both 605.759 Independent Project in Bioinformatics and 605.802.
Prerequisite(s): 605.801 Independent Study in Computer Science I and permission of a faculty mentor, the student's academic advisor, and the program chair.;