Proposed Regular Course Course Descriptions

Application Servers

This course concentrates on architecting, designing, and developing persistent software applications using application server technology. Throughout the course, students are exposed to the evolution of application server architectures that started in the mid-1990s, and experiment with the various architectural styles that are layered on top of mainstream vendor application server platforms today. The course conveys the necessary skills to select the proper application server architectural styles/patterns and platforms based on projects' business and technical requirements. Students will learn how to configure and operate application servers in production environments, taking advantage of features available in mainstream commercial frameworks such as scalability, concurrency, security, fault tolerance, auto-deployment, communications support, development environment, and monitoring tools.

As they design and implement applications using commercial and open source application server technologies (e.g., Spring Framework, IBM WAS, Oracle WebLogic Suite, .Net, OpenCCM), students will learn how to identify architectural and implementation styles and patterns that lead to the most effective use of the various services provided within application server-driven frameworks. Students will also learn how to select and work with related standards driving the implementation and use of the various mainstream application server-driven frameworks (e.g., OMG's OMA/CORBA, JEE, .Net, WS-I) Case studies, provided as part of the course, will demonstrate how medium- to large-size sites manage the complexities inherent in these endeavors. As part of the course, students will also be exposed to next generation application server technologies including Model Driven Architectures (MDAs), Pattern Driven Architectures (PDAs), as well as reflective, multimedia and agent-enabled frameworks.

Bioinformatics

This course introduces the students to Bioinformatics, which uses information technology to store, annotate, retrieve, transfer, integrate and analyze in understanding biological information. Genome-scale sequencing projects have led to an explosion of genetic sequences available for automated analysis. These gene sequences are the codes, which direct the production of proteins that in turn regulate all life processes. The student will be shown how these sequences can lead to a much fuller understanding of many biological processes important to pharmaceutical and biotechnology industries: e.g., to determine for example new drug targets or to predict if particular drugs are applicable to all patients. Students will be introduced to the basic concepts behind Bioinformatics and Computational Biology tools, algorithms and data structures. Hands-on sessions will familiarize students with the details and use of the most commonly used online tools and resources. The course will cover the use of NCBI's Entrez, BLAST, PSI-BLAST, ClustalW, Pfam, PRINTS, BLOCKS, Prosite and the PDB.

Computational Photography

Computational Photography is an exciting new area at the intersection of Computer Graphics and Computer Vision. Through the use of computation, its goal is to move beyond the limitations of conventional photography to produce enhanced and novel imagery of the world around us. The main focus of the course will be on software-based methods for producing visually compelling pictures. However, it will also cover novel camera designs, for which computation is integral to their operation. The course will explain the principles behind many of the advanced tools that can be found in Adobe Photoshop, although the use of this package itself is outside the scope of the course. The course will be suitable for advanced undergraduates, masters and PhD students. A reasonable knowledge of linear algebra is required and familiarity with Matlab is desirable. Assessment will be through coursework and a course project.

Computational Systems Biology

Presently, there is no clear way to determine if the current body of biological facts is sufficient to explain phenomenology. In the biological community, it is not uncommon to assume certain biological problems to have achieved a cognitive finality without rigorous justification. In these particular cases, rigorous mathematical models with automated tools for reasoning, simulation, and computation can be of enormous help to uncover cognitive flaws, qualitative simplification or overly generalized assumptions.

Some ideal candidates for such study would include: prion hypothesis, cell cycle machinery, muscle contractility, processes involved in cancer, signal transduction pathways, circadian rhythms, and many others. We believe that the difficulty of biological modeling will become acute as biologists prepare to understand even more complex systems.
Fortunately, in the past, similar issues had been faced by other disciplines: for instance, design of complex microprocessors involving many millions of transistors, building and controlling a configurable robots involving very high degree-of-freedom actuators, implementing hybrid controllers for high-way traffic or air-traffic, or even reasoning about data traffic on a computer network. The approaches developed by control theorists analyzing stability of a system with feedback, physicists studying asymptotic properties of dynamical systems, computer scientists reasoning about a discrete or hybrid (combining discrete events with continuous events) reactive systems---all have tried to address some aspects of the same problem in a very concrete manner. In this course, biological processes will be studied in a similar manner, as the appropriate tools become available.

Computer Games

You will learn about computer game genres, from casual games to first person 3D games to asynchronous multiplayer games and beyond, and you will learn principles of game design hands-on, by building rapid prototypes of 2D games, followed by larger team projects to design, implement and evaluate 3D. You will take turns assuming different roles within these teams: level design, A.I. programming, character design, animation. There will be an end-of-semester group show, to which the public will be invited, in which you will have an opportunity to demonstrate your best work. etc. games.

Data Mining

Data mining is the application of machine learning techniques for automatic analysis and extraction of useful knowledge from data repositories. The course will introduce concepts and techniques of data mining and data warehousing, including unsupervised and supervised learning, classification, clustering , association rules, decision trees, genetic algorithms, knowledge discovery in databases, OLAP (On-line Analytical Processing), the data warehouse, neural networks, statistical techniques, and rule-based systems.

Prerequisites: G22.1180 Mathematical Techniques for CS Applications and a course on databases (undergraduate or graduate, covering the relational model through optimization and SQL).

Distributed Systems

Distributed systems help programmers aggregate the resource of many networked computers to construct highly available and scalable services. This class teaches the abstractions, design and implementation techniques that allow you to build fast, scalable, fault-tolerant distributed systems. Topics include multithreading, network programming, consistency, naming, fault tolerance, and security and several case studies of distributed systems.

Enterprise Architecture Frameworks

Alignment of business and technology models is a tactical business objective that supports all business strategies meant to improve the value chain in service-driven organizations. The focus of Enterprise Architectures is to facilitate this alignment by identifying the main components of an organization or a sub-set of it (such as its information systems), and the ways in which these components work together. The components include staff, business processes, technology, information, as well as financial and other resources. This course gives an introduction to Enterprise Architecture Frameworks (EAFs) that may be used to catalog and document Enterprise components to help inform, guide, and constrain choices in business/IS/IT solutions development. EAFs are increasingly used in the industry today as a result of the continued emergence of new technologies and ongoing pressures to reengineer business processes to achieve improved efficiency and greater customer focus.

Financial Computing

This course is intended to introduce the students to the basic concepts of Computational Finance and explore various relations between Computer Science and Finance. In particular, the course will introduce both theoretical and practical aspects of finance with an emphasis on the relation between real-life applications and these concepts. We will cover various issues like high-frequency market simulators, framework for performing statistical simulations, we will discuss and model various financial instruments. Strong emphasis will be put on efficiency and proper design. As such, Object Oriented concepts will be discussed and put to use in real life applications. Prerequisites: Fundamental Algorithms, Programming Languages, background in calculus and linear algebra.

Financial Software Projects

The theme of this course will be an "applied case study" and will focus on Fixed Income markets. The semester will begin with a big-picture view of the markets, the inner workings of an investment bank, the market players, and where software engineers fit in. The students will be grouped into small teams to build a financial application using practical software engineering principles."

It is expected that each team will build a risk management framework, starting with basic components; some built by the students and some provided. The professor and guest lecturers will draw on their own experiences in the industry to make this as practical, informative (and entertaining) as possible.

Prerequisites: It is assumed that the students can code in C++. No prior experience in the financial sector domain is required- just a desire to learn.

Networks and Mobile Systems

This is a graduate level class that focuses on networking, communications and systems aspects of mobile devices and mobile computing. The course will begin with a short introduction to core and traditional networking principles before diving into detail about wireless communications and mobile systems. The core networking principles covered will include naming, addressing, routing and transport layer issues beyond which we will cover the following topics: wireless communication principles, wireless MAC layer (802.11 a/b/g/n), signals and communications, cellular networks (2G,3G,4G,WiMax), baseband and wideband concepts, software-defined radios, programmable wireless platforms, next-generation mobile applications and services, mobile cloud computing, mobile operating systems, wireless security, trusted computing platforms for mobile devices, mobile device hardware, power challenges, mobile virtualization, mobile challenges in emerging regions. This is a project oriented Capstone class and all students are expected to do a large project relating to mobile networks and systems.

Production Quality Software

In this course, students learn to develop production quality software. Lectures present real-world development practices that maximize software correctness and minimize development time. A special emphasis is placed on increasing proficiency in a particular programming language by doing weekly development projects and participating in code reviews. Assignments become more sophisticated as the semester progresses, eventually incorporating unit tests, build scripts, design patterns, and other techniques. The course culminates with an assignment that requires students to contribute to an open-source project of their choice.

Random Graphs

Random Graphs, generalized Randomized structures, Random processes, Probabilistic Methods, or Erdos Magic. Branching Processes. Phase transitions for large random evolutions. Derandomization via conditional expectations. Semidefinite programming derandomization techniques. Algorithms, probability and discrete mathematics all appear, but concepts will be defined from scratch. Emphasis will be on the methodologies. Emphasis will be on methods of asymptotic calculation throughout.

Speech Recognition

This course gives a computer science presentation of automatic speech recognition, the problem of transcribing accurately spoken utterances. The description includes the essential algorithms for creating large-scale speech recognition systems. The algorithms and techniques presented are now used in most research and industrial systems.

Many of the learning and search algorithms and techniques currently used in natural language processing, computational biology, and other areas of application of machine learning were originally designed for tackling speech recognition problems. Speech recognition continues to feed computer science with challenging problems, in particular because of the size of the learning and search problems it generates.

The objective of the course is thus not just to familiarize students with particular algorithms used in speech recognition, but rather use that as a basis to explore general text and speech and machine learning algorithms relevant to a variety of other areas in computer science. The course will make use of several software libraries and will study recent research and publications in this area.

This course is also open to undergraduate students.