Course Description

CS 603 Research Methodology

This course aims to study the Nature of research, types of research, research process and its management, ethical issues in research, outlining research problems and developing research questions. Research methods, qualitative and quantitative approaches. Literature surveys and critical analysis and evaluation of sources. Data collection and data analysis (including: survey and questionnaire design, experimental design, use of control groups), Critical evaluation and appraisal of published work and data sets. Recognition of appropriate and inappropriate use of approaches and statistics. Writing a technical paper, the publication process, reviews and the role of the reviewer. The use of peer review. Presenting work orally or by other appropriate means. Structuring a research report. Professional, ethical and legal issues that relate to the relevant program of study.

CS 630 Advanced Operating Systems

This course builds on the foundation material presented in the undergraduate operating system courses. It addresses and explores internals of new operating systems, regarding issues listed in the catalog description. The course is designed to provide students with advanced concepts in operating systems. Topics covered in the course include: Properties of different types of operating systems; Different ways of structuring and designing operating systems; Processes, multithreading, inter-process communication and deadlocks; Memory management and virtual memory systems; File systems; Protection and Security, and advanced operating systems case studies.

In this course, we also take a closer look at the new directions in operating systems research. Students will be asked to read and summarize some research papers in different areas of operating systems.

CS 631 Advanced Computer Architecture

This course focuses on modern advancements in parallel computer architecture, with emphasis on advanced instruction level parallelism (ILP) and multiprocessor architectures. Topics include: advanced branch prediction, data speculation, computation reuse, memory dependence prediction, trace caches, dynamic optimizations, checkpoint architectures, latency-tolerant processors, simultaneous multithreading, speculative multithreading, virtual machines, message passing multiprocessors, UMA, NUMA and COMA shared- memory multiprocessors, single-chip multiprocessors, wormhole routing techniques, cache coherence, memory consistency models, high performance synchronization methods, speculative lock elision and transactional memory. A key component of the course is a research project in which students use architecture performance simulator to investigate novel architecture techniques.

CS 632 Parallel Processing

This course covers the architecture and enabling technologies of parallel computing systems and their innovative applications. We will cover scalable multiprocessors, distributed clusters, computational Grids, cloud computing, virtual machines. The course aims to acquaint Master students in computer science with state-of-the-art supercomputers and distributed computing systems for high-performance computing, grid, cloud, cudascale applications.

CS 634 Computer Networks Architecture

This course aims to provide the needed knowledge to Design and develop an advanced computer communication networks. the topics of this course includes, distributed and failsafe routing in large and dynamic networks, gateways and interconnection of heterogeneous networks, flow control and congestion avoidance techniques, network architectures, communication protocol standards, formal specification and verification of protocols, implementation and conformance testing of protocol standards, network security and web computing.

CS 636 Distributed Systems

This Course is an introductory course in distributed systems. The emphasis will be on the techniques for creating functional, usable, and high-performance distributed systems. To make the issues more concrete, the class includes several multi-week projects requiring significant design and implementation. The goals of this course are twofold: First, students will gain an understanding of the principles and techniques behind the design of distributed systems, such as locking, concurrency, scheduling, and communication across networks. Second, students will gain practical experience in designing, implementing, and debugging real distributed systems. The major themes this course will teach include process distribution and communication, data distribution, scheduling, concurrency, resource sharing, synchronization, naming, abstraction and modularity, failure handling, protection from accidental and malicious harm, distributed programming models, distributed file systems, virtualization, and the use of instrumentation, monitoring and debugging tools in problem solving. As the creation and management of software systems is a fundamental goal of any undergraduate systems course, students will design, implement, and debug large programming projects. Students will learn the design and implementation of today's popular distributed system paradigms, such as Google File System and MapReduce.

CS 638 High Performance Computing

This course aims to provide students with some basic concepts in parallel and high performance computing. The course covers the following topics: Advanced computer architectures, parallel algorithms, parallel languages, and performance-oriented computing.

CS 651 Advanced Algorithms Analysis and Design

This course is an advanced course in algorithms design and analysis. We will look at many new topics and also revisit some you have seen to look at them in more detail. We will likely begin by reviewing sorting and graph algorithms the first week or two. We will proceed to study approximation algorithms, NP-completeness, heuristic algorithms, randomized algorithms. For each algorithm we study, we will focus on verifying that the algorithm even is correct, and also analyze the running time and memory space needed to complete the algorithm. We will choose a number of the algorithms to implement in code. Complexity analysis of algorithms; design and analysis of computer algorithms such as: divide and conquer, greedy method, dynamic method. Backtracking, approximation and randomization applied to polynomial and NP-hard problems.

CS 670 Advanced Artificial Intelligence

This course aims to provide students with some basic concepts in Intelligent agents, deduction, symbolic reasoning, searching, statistical pattern classification, searching cognitive modeling, planning, robotics, vision, machine learning, genetic algorithms, parallel and distributed artificial intelligence.

CS 671 Natural Language Processing

This course aims to provide students with some basic concepts in Natural language processing. The course covers the following topics: Syntax Processing, Semantic Analysis and Strategies, content based analysis and general knowledge, Query and Question answering systems, language generator systems (Example: Machine Translation), and Arabic Language Applications.

CS 672 Knowledge Based Systems

This course aims to provide students with some basic concepts in Knowledge representations and mappings, approaches and issues (e.g. predicate logic, fuzzy logic, week and strong slot and filler structures), knowledge acquisition, the frame problem, symbolic reasoning under uncertainty (nonmonotonic reasoning, augmenting a problem Solver), statistical reasoning (e.g. probability and Bays Theorem, Bayesian networks, Dumpster-Shafer theory), building knowledge-based systems.

CS 675 Machine Learning

Machine learning is the science of algorithms that improve their performance by learning from experience; most often in the form of data with or without labeled examples. Machine learning algorithms are used within a large number of application fields. Independently of the field, a developer of such algorithms need to have a systematic understanding of how a given assignment can be formulated as a machine learning problem. The aim of this course is to give students this systematic understanding. We will present a number of machine learning algorithms and statistical modeling algorithms. But above all, you will learn how the different algorithms are constructed, how they relate to one another and when they are applicable.

CS 681 Image Processing and Computer Vision

This course is an introduction to the exciting and rapidly advancing fields of image processing and computer vision. The vast majority of data on the internet is no longer text; it is audio, image, and video data. Image and video data rates are trending upwards at an enormous pace, so we need to develop fast and accurate techniques to manage and search these non-text data. Moreover, the whole television and film industry is converting to digital format with internet delivery being the preferred method to reach consumers. This course is in two parts. The first part will cover fundamental techniques in image processing.

The second part will cover techniques and applications in video analysis and computer vision. One goal of computer vision is to enable computers to analyze extract high level information from the world from images in a similar way to humans. The course will be delivered in a practical manner and students will be asked to code algorithms in Matlab/ OpenCv library C++ to demonstrate deep and practical understanding of the course material. Image Processing and Computer Vision research strengths will be highlighted including medical imaging and real-time non-cooperative face recognition.

CS 683 Information Security

This course provides a one-semester overview of information security. It is designed to help students with prior computer and programming knowledge to understand this important priority in society today. The technical content of the course gives a broad overview of essential concepts and methods for providing and evaluating security in information processing systems (operating systems and applications, networks, protocols, and so on). In addition to its technical content, the course touches on the importance of management and administration, the place information security holds in overall business risk, social issues such as individual privacy, and the role of public policy. The course will be organized around a few broad themes:

- Foundations: security mindset, essential concepts (policy, CIA, etc.)
- Software security: vulnerabilities and protections, malware, program Analysis
- Practical cryptography: encryption, authentication, hashing, symmetric and asymmetric crypto
- Networks: wired and wireless networks, protocols, attacks andcountermeasures
- Applications and special topics: databases, web apps, privacy and anonymity, voting, public policy

CS 691 Special Topics

The course provides students with varied topics in computer science and technologies, presented as a result of technological change or community or student interest not covered by other CS curricular offerings. Topics vary and are selected from contemporary applications and theory in computer science. Topics may include, but not limited to, (A) Computer Science Theory; (B) Architecture; (C) Programming Languages and Compilers; (D) Operating Systems; (E) Artificial Intelligence; (F) Computer Graphics; (G) Networks; (H) Computer-Aided Design; (I) Scientific Computing; (J) Machine Learning and Optimization; (K) Computer Science.

CS 695 Graduation Project

Students are allocated a supervisor who is a member of academic staff. The supervisor provides a project idea, support and guidance. Student are expected to meet with their supervisor on a regular basis, as agreed with the supervisor schedule. A typical project will comprise approximately 4 weeks of background reading, analysis and design, followed by 4 weeks of implementation and finally 4 weeks to write an in-depth technical report. Exceptionally a student may undertake a project that does not require programming skills. However, a student undertaking such a project would be required to demonstrate an exceptionally high level of skill in the areas of analysis synthesis critical assessment and design. It is not a soft option and students are advised that such a project is typically much harder than a 'programming' project. Some projects are done in conjunction with other departments of the College. Others are done in conjunction with external organizations although supervision is always provided within the department.

CS 698 Comprehensive Exam

Upon completion of 8 core courses (24 credit hours) and 3 elective courses (9 credit hours for a total of 33 completed credit hours; students must successfully complete a 0 credit hours Comprehensive Exam. The Comprehensive Exam assesses general and specific knowledge of computer science. There will be three sessions, where in each one, the student will be assessed with a two hours on-site sit-down comprehensive exam about a computer science pivot. The specification of each pivot is decided by the department. The student should succeed a total point of 140 to pass the exam. The Exam allows faculty to evaluate courses, programs, and the knowledge base gained by individual students during their Masters studies. The Comprehensive Exam also allows faculty to determine that all students have a knowledge base to prepare them to be successful educators.

CS 699 Thesis

A research-based thesis course that offers students the opportunity to work on a comprehensive, individual project that demonstrates mastery of interaction design and/or design computing. Topic to be agreed in consultation with a supervisor. The project will be of suitable complexity for results to be published for an expert audience.