COMPUTER & INFORMATION TECHNOLOGY (EG) {CIT}

L/R 590. Programming Languages and Techniques.. (C)

Introduction to fundamental concepts of programming and computer science for students who have little or no experience in these areas. Principles of modern object-oriented programming languages: abstraction, types, polymorphism, encapsulation, and inheritance. Basic algorithmic techniques and informal complexity analysis. Graphical user interfaces. Substantial programming assignments in Python and Java. This course is for students who do not have an academic background in computer science and who are not pursuing the Master's in Computer Information Technology and who are not graduate students in the CIS Department.

L/R 591. Introduction to Software Development. (C)

Introduction to fundamental concepts of programming and computer science. Principles of modern object-oriented programming languages: abstraction, types, polymorphism, encapsulation, inheritance, and interfaces. This course will also focus on best practices and aspects of software development such as software design, software testing, pair programming, version control, and using IDEs. Substantial programming assignments.

L/R 592. Mathematical Foundations of Computer Science. (C)

This course introduces you to math concepts that form the backbone of the majority of computer science. Topics covered include sets, functions, permutations and combinations, discrete probability, expectation, mathematical Induction and graph theory. The goal of the course is to ensure that students are comfortable enough with the math required for most of the CIS electives. CIS 502 and CIT596 heavily rely on concepts taught in this course.

L/R 593. Introduction to Computer Systems. (C)

This course provides an introduction to fundamental concepts of computer systems and computer architecture. You will learn the C programming language and an instruction set (machine language) as a basis for understanding how computers represent data, process information, and execute programs. The course also focuses on the Unix environment and includes a weekly hands-on lab session.

L/R 594. Data Structures and Sofware Design. (C) Prerequisite(s): CIT 591 or consent of the instructor.

This course will focus on data structures, software design, and advanced Java. The course starts off with an introduction to data structures and basics of the analysis of algorithms. Important data structures covered will include arrays, lists, stacks, queues, trees, hash maps, and graphs. The course will also focus on software design and advanced Java topics such as software architectures, design patterns, networking, multithreading, and graphics. We will use Java for the entire course.

L/R 595. Computer Systems Programming. (C) Prerequisite(s): CIT 593 or equivalent.

This course builds on your knowledge of C and focuses on systems programming for Linux, specifically the libraries that programmers use for threading and concurrency, synchronization, inter-process communication, and networking. The course also introduces the C++ programming language, and covers important concepts in modern operating systems, including processes, scheduling, caching, and virtual memory. After completing this course, you will have the requisite knowledge and experience for systems-focused CIS graduate-level electives.

COMPUTER & INFORMATION TECHNOLOGY (EG) {CIT}

L/R 596. Algorithms and Computation. (C) Prerequisite(s): CIT 592 or equivalent.

This course focuses primarily on the design and analysis of algorithms. We will begin with sorting and searching algorithms and then spend most of the course on graph algorithms. In order to study graph algorithms, general algorithm design patterns like dynamic programming and greedy algorithms will be introduced. A section of this course is also devoted to understanding NP-Completeness.

L/R 597. Programming for the Web. (C) Prerequisite(s): CIT 591 or equivalent and CIT 594 or equivalent. No prior experience with C# or .NET required.

This course will focus on web programming. The first half will focus on the the basics of the Internet and the Web, HTML and CSS, and basic and advanced Ruby. The second half will focus on Rails. Teams (of size 2-3) will build a web application in the second half of the semester as the class project. Through Rails, we'll explore the "culture" of web programming such as agile methodology, testing, key aspects of software engineering, using web services and APIs, and deploying to the cloud.