COURSE DESCRIPTION

Department and Course Number	**CS 425**	Course Coordinator	**Reilly**
Course Title	**Computer System Performance Analysis**	Total Credits	**3**

Current Catalog Description
Computer Systems addressed in this course primarily are web based systems and capacity planning is a principal theme. However, the queueing theory and statistical analysis approaches are applicable to conventional computing systems and, in fact, modeling of these latter constitute relevant background information that is developed and exploited for web systems analysis.

Textbook
Menasce, D. A. and V. A. F. Almeida. Capacity Planning for Web Services: Metrics, Models and Methods. Prentice-Hall, 2002.

References *Occasional reference to information on the internet, particularly related to general purpose (discrete systems) simulation, e.g., the SLX simulation language.*

Course Goals
The main idea behind the course is to gain competence in the central topics of modeling computing systems, particularly, from a mathematical and statistical point of view. In this regard stress is put, primarily on foundations of queueing theory and secondarily on statistical modeling (in a general linear model framework). By this combination the student learns to handle data that may be collected both independent of and dependent on queueing models. For example, estimating future "load" inhabits a key part of the terrain.

Prerequisites by Topic
Algorithms and Data Structures, basic probability and statistics, with software concepts (e.g. lifecycle basics)

Major Topics Covered in the Course
Rigorous specification of queuing system models, development of solutions to several classical queueing models applicable in computer modeling, application of results through use of already prepared and student-developed computations (in spreadsheets and occasionally via simulations). Some attention is made considering related topics, e.g., tools for collecting data about systems.

Laboratory projects (specify number of weeks on each)
Lab work consists primarily in project work, with formal write-ups and oral presentations. Sometimes smaller homework problems are introduced to make specific points.

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
		3			10
Data Structures			Computer Organization and Architecture		
		25			4
Algorithms Software Design			Concepts of Programming Languages		

Oral and Written Communications

Every student is required to submit at least 2 written reports (one of which may be a "warm up") of typically 5-10 pages; "rich" appendices should back up computing results as applicable. One oral presentation of typically 10-20 minutes duration is a goal. Material is graded primarily for technical content and relationship to course material, clarity of presentation (mainly a matter of organization), completeness, and accuracy.

Social and Ethical Issues

Honesty in dealing with collected data and in computational results (i.e., interpretations). Teamwork is allow in project but only for a portion, typically, the first phases; each student then must present some initiatives of his/her own.

Theoretical Content

Probability and statistics – up to 75% of the course. Remarks on algorithms and complexity, programming language theory, software engineering add a few more (theoretical) percentage points.

Problem Analysis

Every problem approached requires analysis - i.e., identifying an appropriate starting point (e.g., from among a few alternatives, say, in "systems models") and clearly defining some"value added" (by the student).

Solution Design

Almost all outside work goes through a complete life cycle of analysis, design, implementation and testing.