COURSE DESCRIPTION

Dept., Number	CSci 405	Course Title	Computer Simulation
Semester hours	3	Course Coordinator	Stephen V. Rice, Assistant Professor

Current Catalog Description

Introduction to computer-based simulation and its applications to engineering, the sciences, and management.

Textbook

Averill M. Law. *Simulation Modeling and Analysis*, 4th edition, McGraw-Hill, 2007.

References

Course Outcomes

This course introduces the tools and techniques for writing computer programs that model and simulate systems. The focus is on discrete-event simulation, in which stochastic systems are modeled by discrete changes over time. First the students learn to implement discrete-event simulation programs using a general-purpose programming language, such as C, C++, or Java. Then the students learn to use two simulation programming languages, SIMSCRIPT and GPSS. After successfully completing this course, students are able to:

1. identify and model relevant characteristics and behaviors of system entities
2. understand the internals of a discrete-event simulation engine
3. select appropriate probability distributions for random-number generation
4. determine and collect meaningful statistics of system performance

Relationship between Course Outcomes and Program Outcomes

The course outcomes contribute to the program outcomes as follows: (1) to (c), (2) to (i), (3) to (j), (4) to (c).

Prerequisites by Topic

1. Fundamental algorithms and data structures (CSci 112, 211)
2. Intermediate programming concepts and skills (CSci 211)
3. Introductory differential and integral calculus (Math 262)

Major Topics Covered in the Course

- Discrete events, the simulation clock, the event set, event routines, and the timing routine
- Simulation statistics, including time-weighted statistics
- Probability distributions and random-number generation
- SIMSCRIPT simulation programming language and process-oriented simulation
- GPSS simulation programming language

Assessment Plan for the Course

This is an elective course offered approximately every two years. An offering typically has three examinations and four programming assignments (including a large simulation project), which are designed to assess course outcomes (1) to (4).

How Data in the Course are Used to Assess Program Outcomes (unless adequately covered already in the assessment discussion under Criterion 4)

Estimate Curriculum Category Content (Semester hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms			Software design		1.25
Data structures		0.50	Concepts of programming languages		1.25