



King Saud University
College of Computer and Information Sciences
Department of Computer Science

CSC 394 – Selected Topics in Computer Science (3-0-1) - Elective Course

Catalog description:

1. Robust communication software and architectures:
Critical systems; Real-time systems; System models; Software architecting; Design patterns for robust software; High-availability; Performance benchmarking.
Next Generation Networks as example of robust communication architectures.
2. Research methodology in Software Engineering; Scientific writing.

Prerequisite:

Prerequisite to:

Co-requisite to:

Texts:

- Greg Utas, *Robust communications software*, Wiley, 2005.
- Ian Sommerville, *Software Engineering*, Addison Wiley, 2008
- Jeong Yun Kim, *NGN architecture and framework*, ETRI white paper, 2008.
- Mary Shaw, *What makes good research in software engineering*, International Journal of Software Tools for Technology Transfer, 2002, vol. 4, no. 1, pp. 1-7.

Course Objectives:

The objective of this course is to develop the students' ability to use the common critical software and architecture patterns. The students learn the characteristic features of critical and real-time systems through both the lectures and a sequence of illustrative programming assignments. Students will also study the NGN standard architecture as example of critical system architecture. Another goal of this course is to ameliorate the students' research and writing skills through lectures and assignments.

Course Outcomes:

Students are expected to be able to:

1. Understand the characteristics of critical and real-time systems
2. Hands-on experience in architecture, protocol and API design using UML
3. Higher design and programming skills using design patterns
4. Scientific research and writing skills

Expected Performance Criteria:

Students should complete case study homework assignments and pass written examinations on class material

Topics:

1. Introduction to software and systems
2. Critical and real-time systems
3. System models
4. Robust software architectures
5. Design patterns
4. High-availability and benchmarking
6. Case study

Schedule:

15 weeks of three 50-minute lectures and one 50-minute tutorial per week.

Relationship of Course to ABET Criteria:

Criterion 2 - Program Educational Objectives:

This course allows the student to gain the necessary skills to lead, design, develop, and maintain computer-related projects in various fields (PEO 1).

Criterion 3 - Program Outcomes:

<i>a. An ability to apply knowledge of computing and mathematics appropriate to the discipline;</i>	

<i>b. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;</i>	

<i>c. An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;</i>	✓
Students design and implement real-time protocols and APIs. They also conduct performance and load tests.	
<i>d. An ability to function effectively on teams to accomplish a common goal;</i>	✓
Students work in teams to realize a project. Also, they practice code peer-reviewing	
<i>e. An understanding of professional, ethical, legal, security, and social issues and responsibilities;</i>	

<i>f. An ability to communicate effectively with a range of audiences;</i>	✓
Students learn how to write scientific reports	
<i>g. An ability to analyze the local and global impact of computing on individuals, organizations and society;</i>	

<i>h. Recognition of the need for, and an ability to engage in, continuing professional development;</i>	✓
Students learn to apply appropriate research methods to solve problems	
<i>i. An ability to use current techniques, skills, and tools necessary for computing practices.</i>	✓
Students learn and apply well known and proved design patterns	

<i>j. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;</i>	✓
Students design and implement real-time protocols and APIs. They also conduct performance and load tests. They update their designs in order ameliorate their programs' performance.	
<i>k. An ability to apply design and development principles in the construction of software systems of varying complexity.</i>	✓
Students design, implement and test a number of programming projects	

Criterion 4 – Professional Component:

The course prepares students to be effective software design and development professionals.

Prepared by: Zohair Chentouf, October 4, 2010

Reviewed by:

Revised by:

Approved by: