

**UNIVERSITY OF NEBRASKA AT OMAHA  
COURSE SYLLABUS/DESCRIPTION**

Department and Course Number	CSCI 4830
Course Title	Introduction to Software Engineering
Course Coordinator	Harvey Siy
Total Credits	3
Date of Last Revision	November 20, 2008

**1.0 Course Description**

- 1.1 Overview of content and purpose of the course (Catalog description).  
Basic concepts and major issues of software engineering, current tools and techniques providing a basis for analyzing, designing, developing, maintaining and evaluating software systems. Technical, administrative and operating issues. Privacy, security and legal issues.
- 1.2 For whom course is intended.  
The course is intended for upper division undergraduate CS or MIS majors who wish to pursue the topic of Engineering and development of software systems.
- 1.3 Prerequisites of the course (Courses).  
CSCI-3320
- 1.4 Prerequisites of the course (Topics).
  - 1.4.1 Concepts of structured problem solving and programming
  - 1.4.2 Concepts of object-oriented problem solving programming
  - 1.4.3 Familiar with advanced data structures
  - 1.4.4 Familiar with main concepts of analysis of algorithms
  - 1.4.5 Proficiency in one of the modern programming languages
- 1.5 Unusual circumstances of the course.  
None.

**2.0 Objectives**

- 2.1 List of performance objectives stated in terms of the student educational outcomes.
    - 2.1.1 Perform analysis and design of small and medium-sized software project using structured methods.
    - 2.1.2 Be able to participate in design of small and medium-sized software project using object-oriented software development methodologies.
    - 2.1.3 Prepare software project management documents.
    - 2.1.4 Be able to participate in a project team.
    - 2.1.5 Develop parts/whole prototype as well as implementation of small or medium-sized software projects.
    - 2.1.6 Introduce socio-technical and ethical issues in the development of real-world software systems.
- This course covers topics on development of software systems. It provides students with knowledge of performing system and software requirement analysis and specification,

architecture and detailed design, testing, and integration techniques. It also presents the basics of project management and object oriented methodologies

Software project management

Software life cycle and process

Requirement analysis

System and information engineering

Analysis and design methods

### **3.0 Content and Organization**

List of major topics to be covered in chronological sequence .

- 3.1 Introduction (1.5)
  - 3.1.1 FAQs about software engineering
  - 3.1.2 Professional and ethical responsibility
- 3.2 Computer-based System Engineering\* (3)
  - 3.2.1 Emergent system properties
  - 3.2.2 Systems and their environment
  - 3.2.3 System modeling
  - 3.2.4 The system engineering process
  - 3.2.5 System procurement
- 3.3 Software Processes (2.5)
  - 3.3.1 Software process models
  - 3.3.2 Process iteration
  - 3.3.3 Software specification
  - 3.3.4 Software design and implementation
  - 3.3.5 Software validation
  - 3.3.6 Software evolution
  - 3.3.7 Automated process support
- 3.4 Project Management (2.5)
  - 3.4.1 Management activities
  - 3.4.2 Project planning
  - 3.4.3 Project scheduling
  - 3.4.4 Risk management
- 3.5 Managing People\* (1.5)
  - 3.5.1 Limits to thinking
  - 3.5.2 Group working
  - 3.5.3 Choosing and keeping people
  - 3.5.4 The people capability maturity model
- 3.6 Software Cost Estimation\* (2.5)
  - 3.6.1 Productivity
  - 3.6.2 Estimation techniques
  - 3.6.3 Algorithmic cost modeling
  - 3.6.4 Project duration and staffing
- 3.7 Quality Management\* (1.5)
  - 3.7.1 Quality assurance and standards
  - 3.7.2 Quality planning
  - 3.7.3 Quality control

3.7.4	Software measurement and metrics	
3.8	Process Improvement*	( 1.5)
3.8.1	Process and product quality	
3.8.2	Process analysis and modeling	
3.8.3	Process measurement	
3.8.4	The SEI process capability maturity model	
3.8.5	Process classification	
3.9	Software Requirements	( 2.5)
3.9.1	Functional and non-functional requirements	
3.9.2	User requirements	
3.9.3	System requirements	
3.9.4	The software requirements document	
3.10	Requirements Engineering Processes*	( 1.5)
3.10.1	Feasibility studies	
3.10.2	Requirements elicitation and analysis	
3.10.3	Requirements validation	
3.10.4	Requirements management	
3.11	System Models	( 6)
3.11.1	Context models	
3.11.2	Behavioral models	
3.11.3	Data models	
3.11.4	Object models	
3.11.5	CASE workbenches	
3.12	Software Prototyping*	( 1)
3.12.1	Prototyping in the software process	
3.12.2	Rapid prototyping techniques	
3.12.3	User interface prototyping	
3.13	Architectural Design*	( 3)
3.13.1	System structuring	
3.13.2	Control models	
3.13.3	Modular decomposition	
3.13.4	Domain-specific architectures	
3.14	Object-oriented Design*	( 3)
3.14.1	Objects and object classes	
3.14.2	An object-oriented design process	
3.14.3	Design evolution	
3.15	Design with Reuse*	(1)
3.15.1	Component-based development	
3.15.2	Application families	
3.15.3	Design patterns	
3.16	User Interface Design*	( 1.5)
3.16.1	User interface design principles	
3.16.2	User interaction	
3.16.3	Information presentation	
3.16.4	User support	
3.16.5	Interface evaluation	