UNIVERSITY OF NEBRASKA AT OMAHA
COURSE SYLLABUS

Department and Course Number	CSCI 4900
Course Title	*Internet Systems Development*
Course Coordinator	*Harvey Siy*
Total Credits	*3*
Repeat for Credit?	*No*
Date of Last Revision	*March 3, 2015*

1.0 Course Description Information

1.1 Catalog description:

This course focuses on contemporary techniques and technologies in the design, development, and integration of web-enabled information systems. Topics include: Multi-tiered systems architecture; rapid application development; object-oriented analysis and design; prototyping; testing, verification, and validation; lifecycle models; and component-based development. This is a rapidly moving, hands-on course that mirrors real-world development of internet-based applications.

1.2 Prerequisites of the course:

1.2.1 CSCI 1620 (Introduction to Computer Science II) or proficiency with Java
1.2.2 CSCI 2850 (Programming on the Internet) or introductory experience creating simple web-based applications using server-side technologies such as PHP, CGI, ASP, or Cold Fusion
1.2.3 (recommended) CSCI 3830 (Advanced Java Programming)
1.2.4 CSCI 4830 (Introduction to Software Engineering) or equivalent
1.2.5 Experience with working on a software engineering team project
1.2.6 General understanding of networking concepts such as OSI reference model and TCP/IP
1.2.7 Completion of two elective courses within the iT concentration

1.3 Overview of content and purpose of the course:

This course acts as the capstone course for the Internet Technologies (iT) concentration. Students apply their knowledge of internet technologies and software engineering principles to develop a non-trivial web-enabled information system using an iterative process model. The course also introduces students to modern software architectures as well as software engineering technologies geared towards component-based web application development.

1.4 Unusual circumstances of the course.

none

2.0 Course Justification Information

2.1 Anticipated audience / demand:

Students pursing either a BSMIS or BSCS degree with the Internet Technologies (iT) concentration who have achieved senior standing.

2.2 Indicate how often this course will be offered and the anticipated enrollment:

Yearly

2.3 If it is a significant change to an existing course, please explain why it is needed:

n/a

3.0 List of performance objectives stated in learning outcomes in a student's perspective:

3.1 Manage change in the context of distributed systems development & maintenance;
3.2 Define appropriate architectures for distributed applications systems and the infrastructure supporting them;
3.3 Apply appropriate methodologies for distributed systems development;
3.4 Conduct analysis, design, and implementation using an object-oriented paradigm; and
3.5 Create Internet-based, distributed systems.

4.0 Content and Organization Information

4.1 List the major topics central to this course:

- 4.1.1 Introduction
  - 4.1.1.1 Introduction : Speed, change, and integration: the impact of current trends on systems development.
  - 4.1.1.2 Technical Architectures
  - 4.1.1.3 Multi-tiered Architecture
  - 4.1.1.4 Client-server systems
  - 4.1.1.5 Internet-based architecture
  - 4.1.1.6 Object-based architecture
- 4.1.2 Middleware
  - 4.1.2.1 Distributed Services
  - 4.1.2.2 Integration
  - 4.1.2.3 Web services
  - 4.1.2.4 Application Servers
- 4.1.3 Methodology
  - 4.1.3.1 Scrum
  - 4.1.3.2 Other agile methodologies
- 4.1.4 Object-oriented technologies
  - 4.1.4.1 Object-oriented analysis
  - 4.1.4.2 Object-oriented design
  - 4.1.4.3 Unified Modeling Language
  - 4.1.4.4 Use cases
  - 4.1.4.5 Test-driven development
- 4.1.5 Modern Enabling Technologies
  - 4.1.5.1 Java EE
  - 4.1.5.2 Web Services & Service Oriented Architectures
  - 4.1.5.3 Mobile Computing
  - 4.1.5.4 Component Models
  - 4.1.5.5 Web Frameworks
  - 4.1.5.6 Message-oriented Middleware
  - 4.1.5.7 UI Frameworks
  - 4.1.5.8 Persistence Frameworks

5.0 Teaching Methodology Information

5.1 Methods:

The course will employ a variety of teaching methods including lectures and hands-on guided projects. Assignments will include a combination of individual and group components. The course may be team taught by two or more professors.

5.2 Student role:

The student will attend lectures, participate in discussions, complete programming assignments, evaluate a web programming framework (e.g. Struts, Spring, etc) and complete both individual and group elements of a semester project.

6.0 Evaluation Information

6.1 Describe the typical types of student projects that will be the basis for evaluating student performance:

Students will work on a capstone systems development project throughout the semester. The project will be broken down into a number of modules that reflect the topics covered in the course, but will also draw on material learned in the prerequisite courses.

6.2 Describe the typical basis for determining the final grade (e.g. weighting of various student projects):

Component	Grading
Exams	20%
Homework	30%
Final Exam	20%
Project	30%

6.3 Grading type:

Percent	***Grade***	***Percent***	***Grade***
97 – 100	*A+*	*76 – 78*	*C+*
92 – 96	*A*	*72 – 75*	*C*
89 – 91	*A–*	*69 – 71*	*C–*
86 – 88	*B+*	*66 – 68*	*D+*
82 – 85	*B*	*62 – 65*	*D*
79 – 81	*B–*	*59 – 61*	*D–*
		0—58	*F*

7.0 Resource Material Information

7.1 Textbooks and/or other required readings used in course:

B. Basham, K. Sierra, B. Bates. Head First Servlets and JSP. O'Reilly Media, 2008.

This book is anchored in the Sun Certified Web Component Developer exam which emphasizes core concepts of Internet systems development and Servlet/JSP containers.

Additionally, students will be assigned readings drawn from the current bibliography or similar sources.

A. Fox and D. Patterson. Engineering Software as a Service: An Agile Approach Using Cloud Computing. Strawberry Canyon, LLC, 2014.

7.2 Other student suggested reading materials:

None

7.3 Current bibliography and other resources:

Students will read industry white papers, tutorials, and on-line discussion forums on subjects related to this course.

7.3.1 Ashmore, D. (2014) “The Java EE Architect's Handbook, Second Edition: How to be a successful application architect for Java EE applications”, DVT Press.
7.3.2 Arlow, J., Neustadt, I. (2001). UML and the unified process: Practical object-oriented analysis and Design. Addison Wesley.
7.3.3 Booch, G., Maksimchuk, R., Engle, M. (2007) “Object-Oriented Analysis and Design with Applications (3rd Edition)”, Addison-Wesley.
7.3.4 Cusumano, M. and Yoffie. D. (1998) Competing on Internet Time: Lessons from Netscape and Its Battle with Microsoft. Free Press/Simon & Schuster, New York.
7.3.5 Cusumano, M. and Yoffie. D. (1999) “Software Development on Internet Time,” Internet Computing, IEEE, Piscataway: NJ. Vol. 32, No. 10; OCTOBER 1999, pp. 60-69
7.3.6 Fowler, M. (2002) “Patterns of Enterprise Application Architecture”, Addison-Wesley.
7.3.7 Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994) “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley.
7.3.8 Guide to the Software Engineering Body of Knowledge SWEBOK (2004), Executive Editors: Alain Abran and James W. Moore, Pierre Bourque and Robert Dupuis. IEEE, Piscataway: NJ.
7.3.9 Läufer, Konstantin, “A Hike through Post-EJB J2EE Web Application Architecture” Computing in Science & Engineering Sept/Oct 2005. On 10/13/08 online at http://tinyurl.com/5xyo6y (which redirects to computer.org)
7.3.10 Szyperski, C (2002) “Component Software: Beyond Object-Oriented Programming, 2nd Ed.,” Addison-Wesley, Harlow: England.
7.3.11 Yener, M., Teedom, A., Rahman, R. (2015) “Professional Java EE Design Patterns”, Wrox.

8.0 Other Information:

8.1 Accommodations statement:

8.2 Other:

8.3 Author(s):

Zachary Fowler, Harvey Siy

9.0 Computer Science Accreditation Board (CSAB) Category Content (class time in hours):

CSAB Category	*Core*	*Advanced*
Data structures		
Computer organization and architecture		6
Algorithms and software design		12
Concepts of programming languages		6

10.0 Oral and Written Communications:

Every student is required to submit at least 1 written reports (not including exams, tests, quizzes, or commented programs) to typically 6 pages and to make 1 oral presentations of typically 15

minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

11.0 Social and Ethical Issues:

No coverage

12.0 Theoretical content:
Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

	Contact Hours
12.1 Concepts of applications and infrastructure architecture, including physical and logical architecture, and component-based construction of software systems	6.0

13.0 Problem analysis:
Please describe the analysis experiences common to all course sections.

Students will learn to conduct analysis using an object-oriented paradigm.

14.0 Solution design:
Please describe the design experiences common to all course sections.

Students will learn contemporary techniques and technologies in the design, development, and integration of web-enabled information systems. Specifically, students will learn to define appropriate architectures for distributed applications systems and the infrastructure supporting them, conduct design, and implementation using an object-oriented paradigm; and they will create Internet-based, distributed systems.

CHANGE HISTORY

Date	***Change***	***By whom***	***Comments***
4/30/2003	In the methodologies unit, replace JAD, RAD, prototyping, and system life-cycle with agile methodologies, particularly extreme programming.	Wolcott	JAD, RAD, prototyping, and system life-cycle are covered adequately in ISQA 4110 and ISQA 4120.
6/25/03	ABET clean-up	Wolcott	
10/13/08	Added homework, presentation, and textbook. Replaced ORBs with SOA.	Payne	Currently, web services and SOA are more common than ORBs in the field.
12/10/08	Inserted the mapping table between program outcomes and course objectives	Siy	
6/16/2011	Updated prerequisites and bibliography	Siy	Added CSCI 3830 and 4830 as prerequisites to reduce duplication of content.
7/29/2011	Updated format to standard template	Siy	
3/3/2015	Updated references	Siy	

S – Strong relationship
X – Contributing relationship

Course objective	(a) knowledge of discipline	(b) analyze problem, define requirements	(c) design and implement solution	(d) function on a team	(e) ethical issues	(f) communicate effectively	(g) analyze impact of computing	(h) continued professioanl development	(i) Current techniques and tools	(j) apply foundations	(k) apply design and development principles