National University
School of Engineering and Computing
Department of Computer Science and Information Systems

Master of Science in Computer Science
CSC686, CSC687: Computer Science Project I, II
May. 2018 – July. 2018

COURSE OUTLINE

CSC686, CSC687: Computer Science Project I, II
Since CSC 686 and CSC687 together constitute a capstone project course sequence in the MS in Computer science program, their outlines are presented together below in order to clarify related issues.

CSC686 Course Description:
A study of the software development practices. It emphasizes logical organization of systems and communicating design through documentation suitable for generating a concrete implementation. Students construct an original project with practical applications utilizing software engineering concepts. This project includes requirements engineering, design, test plans, and user documentation. Grading is by "H" (for Honors, "B" or better work), "S" (for Marginal, "C" level work), or "U" (Unsatisfactory, "D" or below).

CSC687 Course Description:
This is a continuation of the student project from CSC686. Student teams complete the project in this phase. The project is coded, module-tested, system-tested, and all documentation is completed. Grading is by "H" (for Honors, "B" or better work), "S" (for Marginal, "C" level work), or "U" (Unsatisfactory, "D" or below).

Instructors:

Bhaskar Raj Sinha, Ph.D., M.B.A.
Professor, School of Engineering and Computing
National University, 3678 Aero Court, San Diego, CA 92123-1788
Tel. (858)309-3431, Fax (858)309-3419, bsinha@nu.edu

P. Peter Dey, Ph.D.
Professor, School of Engineering and Computing
National University, 3678 Aero Court, San Diego, CA 92123-1788
Tel. (858)309-3421, Fax (858)309-3420, pdey@nu.edu

Pre-requisites: All core MSCS courses or permission of the Lead Faculty [Dr. M. F. Wyne]

Class Schedule: Tuesdays and Thursdays **6:00PM-8:00PM** online on Blackboard

Course Competencies: The overall purpose of this course is to provide a capstone to the computer science degree by requiring the development of a significant software project using software engineering principles and

skills presented throughout the curriculum. Students will participate in a team environment for software development by specifying, designing, coding and testing a significant software program.

Learning Outcomes: Upon successful completion of the course, students will be able to:

- Create software requirements specifications, and design and develop complex software systems using software engineering processes and tools.
- Evaluate computer security vulnerabilities and threats, and countermeasures that are both effective and ethical.
- Analyze, design and develop database solutions by translating database modeling theory into sound database design and implementation.
- Analyze and design complex front-end applications for cloud and client-server architectures and integrate them with backend databases.
- Compare and contrast alternative systems for process and memory management.
- Demonstrate ability to conduct in-depth research, both individually as well as in teams, in a specific computer science area and ability to maintain currency in computer science through lifelong learning.
- Demonstrate critical thinking and ability to analyze and synthesize computer science concepts and skills with ethical standards through graduate-level evaluative and creative written assignments and oral reports.

Material covered in this course also supports the following Learning Outcomes:

- Analyze a computational problem and produce a requirements analysis specification of the problem
- Assess the difference between problem models and solution models
- Given a problem, develop a use-case analysis of the problem
- Apply computational principles such as abstraction, encapsulation, localization to real world problem
- Apply programming models such as Object Oriented Programming, Structured Programming to develop a programming solution
- Utilize design tools for designing a computational system
- Apply case tools to develop a computational solution to a problem
- Develop an application based on a given design and requirements
- Evaluate logical and physical database architecture components, such as concurrency control, query processing, transaction management, and data storage/access.
- Demonstrate database administration and management skills.
- Evaluate design methodologies for software projects including hardware/software integration, networking and graphical user interfaces.
- Develop and evaluate comprehensive software test plans at both the software module and the systems level.
- Analyze and manage software configuration to ensure conformance to Total Quality Assurance standards.
- Design and implement a user interface for a software system in order to maximize its usability.
- Construct a computational model for a given problem and examine its consequences.
- Evaluate programming models such as Object Oriented programming and structured programming.

Class Requirements: Students are expected to participate in all class activities, complete quizzes, and turn in all assignments on time. This capstone course is a relatively independent project that a student completes on his/her own or with other students in a team. In a team project students must participate significantly in all aspects of the team effort. Failure to do so may result in the loss of points. As an option for completing the project, students are allowed to work as an intern and solve a real world problem in cooperation with partners in industry. This option is an integral part of the curriculum and is available to all students in the program.

Student Evaluations: As this course is part of a two-course capstone sequence, the grading is by "H" (numeric score of 84% or better), "S" (74% to 83%) or "U" (73% or below).

Letter grades designated as "A," "B," "C," "D," or "F" will be available for tuition reimbursement purposes only and are not included in the student's overall grade point average.

Grades will be awarded to project teams. Project teams will be made up of one to four students. Normally, all team members will get the same grade. However, members who evidently contribute more or less will get higher or lower grades than the team in general. Absence from scheduled sessions is taken as evidence that the student is not contributing up to standard and may result in the reduction of the grade.

The course grade is based on the development of project documentation, presentations about the project, and a demonstration of the project itself. Project teams are required to prepare and maintain a project schedule and log. The log will be used to report the number of hours sent on the component activities of the project.

Instructional Methods:

Practicum in Computer Science, Software Engineering and Project Management. Lectures. Exercises. Teamwork in Project Development. Presentations by students. World Wide Web research. There is no midterm or final exam.

Student Responsibilities and Scores:

Class	Deliverable	Week	Max Score
CSC686	• Title page: Project name, team members, course name/number, etc. • Project Description • Research and references (Introduction)	Week 1	5
	Project Proposal (Oral and report)	Week 1	5
	Requirements Analysis/Feasibility	Week 2	5
	Project Plan (MS Project or similar)	Week 2	5
	Operational Scenario Templates	Week 2	10
	Use Case Diagrams	Week 2	10
	Activity Diagrams	Week 2	10
	Relational Schema and ERD	Week 3	10
	UI and Transitions	Week3	5
	Test Specification/Test Plan	Week 3	5
	Prototype of key elements in the project	Week 4	10
	Project Report: All above and references	Week 4	10
	Project Defense: Presentation and Review	Week 4	10
CSC687	Overall Project Reviews/Walkthroughs (Oral)	Week 1	5
	Design, Design Review (Discussion), Prototype	Week 2	5
	Project Abstract for Flyer	Week 2	5
	Prototype Demonstrations/Reviews	Week 3/4	5
	Prototype Demonstrations/Reviews	Week 5	5
	Project Report	Week 6	20
	Formal Project Presentation	Week 7/8	50
	Answer to Questions	Week 7/8	5

Project teams are required to prepare and maintain a project schedule and log. The log will be used to report the number of hours spent on the component activities of the project. A weekly status report (refer to handout) is to be submitted to the instructor.

Software Engineering Resources:

Software Engineering Institutions and Associations

- Association for Computing Machinery (ACM). Founded in 1947, ACM is the world's first educational and scientific computing society with members - over 80,000 computing professionals and students world-wide. A good resource for authoritative publications, pioneering conferences, and visionary leadership for the new millennium. http://www.acm.org/
- IEEE Computer Society. With nearly 200,000 members, the IEEE Computer Society is the world's leading organization of computer professionals. http://www.computer.org/
- IEE Informatics pages are designed to be an electronic one-stop-shop for all IEE services relevant to the Computer Systems Professional. http://www.iee.org/oncomms/sector/computing/
- Software Engineering Institute (SEI).The Software Engineering Institute is a federally funded research and development center established in 1984 by the U.S. Department of Defense with a broad charter to address the transition of software engineering technology. http://www.sei.cmu.edu/

- JavaWorld. An excellent source of articles relating to Java and object-oriented programming issues. http://www.javaworld.com
- The Software Engineering Association (SEA) is an informal grouping of academic and practitioners working in the area of software engineering. SEA exists to promote and foster software engineering research, practice and education in the UK and internationally. http://uksoft.co.umist.ac.uk/
- SEWORLD is a mailing list provided as a service to the Software Engineering community, and is intended primarily for the dissemination of time-sensitive information relevant to the field of software engineering research. http://www.cs.colorado.edu/serl/seworld
- Software Engineering Code of Ethics and Professional Practice. http://computer.org/tab/SWECP9912.htm
- Software Engineering Body of Knowledge (SWEBOK). The SWEBOK project is the most important event in the history of software engineering for 30 years. The SWEBOK project provides a consensually validated characterization of the bounds of the software engineering discipline and provides a topical access to the Body of Knowledge supporting the discipline. http://www.swebok.org/
- Centre for National Software Studies (CNSS). The mission of the Centre for National Software Studies is to elevate software to the national agenda, and to provide objective expertise, studies, and recommendations on national software issues. http://www.cnsoftware.org/
- Useful links to Safety-Critical Systems Development: http://www.cs.queensu.ca/Software-Engineering/toolcat.html

Software Engineering Tools and OO Links

- This site provides a comprehensive list of CASE tools by category http://www.cs.queensu.ca/Software-Engineering/toolcat.html
- Software engineering tools - IBM Corp: http://www.research.ibm.com/compsci/plansoft/tools.html
- More than 18,000 links on Objects & Components: http://www.cetus-links.org/

NU Library Electronic Resources:

The NU Library System (NULS) purchases access to several databases of full text articles from scholarly journals. Go to http://www.nu.edu/library and click on "Electronic Resources".