WILLIAM PATERSON UNIVERSITY
College of Science and Health
Department of Computer Science

COURSE OUTLINE

1. TITLE OF COURSE AND COURSE NUMBER:

Software Engineering, CS 3500
Credits: 3 (Technology Intensive and Writing Intensive)

2. DESCRIPTION OF THE COURSE:

The course provides a hands-on experience with the issues and techniques of software engineering. A team project applying the techniques covered is the main focus of the course. This course introduces the fundamental principles and practices of software development process to produce quality software systems. Several development paradigms, processes, models, and methods will be discussed. The topics cover the entire software lifecycle that include requirement analysis and specification, design, implementation, testing, integration, maintenance/evolution, documentation, and project management. The course also introduces APIs, CASE tools and environments, as well as the UML (Unified Modeling Language). This course is both writing and technology intensive.

3. COURSE PREREQUISITES:

CS 2550 (Foundations of Information Systems) or CS 3420 (Data Structures) with a grade of C- or better

4. COURSE OBJECTIVES

The main objectives are:
- To gain experience with all the phases of software lifecycle and understand the concept of software development process and software management.
- To understand that software development as a process that can be measured and improved.
- To identify approaches and standards for software quality enhancement.
- To learn software development beyond that typical encountered in programming courses.
- To develop an understanding of modeling tool and language (UML) for software analysis and design, especially object-oriented techniques.
- To experience working as part of a team.
- To produce project documentation that covers software requirements and specification, design, and implementation.

5. STUDENT LEARNING OUTCOMES
COURSE-SPECIFIC SLOS:

Upon completion of the course, students will be able to:
- Understand the phases of software lifecycle of requirements, design, implementation, testing, installation, and operation and maintenance (T1).
- Explain software development process and object-oriented methodology (T1).
- Perform analysis of a problem to determine user needs and develop specifications of the system required that will meet user needs within economic constraints (W1, W3, T2).
- Perform design from specification and generate codes from design using UML (T1, T2).
- Identify approaches and standards for software quality improvement (T2).
- Gain general concepts of software project management (T1, T3).
- Understand the legal and ethical behaviors in the software industry, as well as the professional code of conduct of a software engineer (T4).
- Produce a project documentation that consists of four major parts of the software development process: requirements, specification, design, and implementation (W1, W2, W3).

Through classroom presentations, participation, discussions, homework, team project, and other assignments, this course also reinforces the following student learning outcomes:
- Communicate effectively through speaking and writing skills (W1, W2, W3).
- Demonstrate understanding of scientific principles and methods (W3, T1).
- Formulate strategies to locate, evaluate, and apply information (W1, W2, T2).
- Identify activities that fulfill personal, civic, and social responsibilities (T3, T4).
- Use computer and emerging digital technologies effectively (T1, T2).
- Demonstrate an awareness of global connections and interdependencies (e.g., collaborative software engineering) (T3).

UCC AREA SLOS: N/A

WRITING INTENSIVE SLOS:

Students will be able to:
W1. Use writing-to-learn strategies (such as brainstorming, free-writing, reading logs, etc.) to develop their understanding of course content and to think critically about that content.
W2. Use drafting, revising, editing and other writing processes to develop final writing products appropriate to the discipline, such as thesis-driven essays, formal reports, or professionally formatted manuscripts.
W3. Use research and documentation skills where they may be necessary and integrate them through paraphrase, quotation and citation, in accordance with the conventions of the discipline.

TECHNOLOGY INTENSIVE SLOS:

Students will be able to:
T1. Demonstrate a sound understanding of technology concepts, systems and operations.
T2. Use a variety of technologies to access, evaluate, collect, and manage data, information and datasets.
T3. Understand the impact of technology on themselves, their culture, their environment and their society.
T4. Practice legal and ethical behaviors in the context of technology.

6. **TOPICAL OUTLINE OF THE COURSE CONTENT**

- Overview of software engineering and its history.
- Software development phases and their artifacts:
  - Requirements and specifications.
  - Design: coupling, cohesion, object orientation.
  - Testing: unit, integration, system/product, and regression testing.
  - Implementation and programming techniques.
  - Delivery and Installation.
  - Maintenance: corrective, adaptive, perfective, preventive.
- Software quality and metrics.
- Object-oriented model and object-oriented concepts of encapsulation, data abstraction, information hiding, inheritance, polymorphism.
- Introduction to CASE tools.
- Software project management and planning, including team organization.

7. **GUIDELINES/SUGGESTIONS FOR TEACHING METHODS AND STUDENT LEARNING ACTIVITIES:**

- Classroom lectures and presentations.
- Student presentations and discussions.
- Classroom hands-on exercises and problem solving sessions.
- Students conduct teamwork sessions outside classroom for software project.

8. **GUIDELINES/SUGGESTIONS FOR METHODS OF STUDENT EVALUATION**

- Weekly homework sets (individual work).
- Group projects (team work): documents of requirements and specifications, design, implementation, and a final project report.
- Team presentation.
- Two examinations and a comprehensive final examination.
Writing Intensive SLO Assessment:
W1. The class will be divided into several 3-person teams. If there are one or two persons left, then one 4-person team or two 4-person teams will be formed. Each team needs to document the four major tasks of software development in three separated documents and a final project report. The concepts and methodologies will be reviewed in homework and tested in the examinations. The homework is about 17% of the final grade. The project contributes to 20% of the final grade. Two exams are totaled 40% and the final comprehensive exam is 23% of the final grade.

W2. Each project team needs to produce three documents during the semester. Each of these documents will be reviewed by the instructor and feedback will be given to the project teams. Based on the feedbacks, each team requires producing a final project report at the end of the semester which consists of the revised documents of the three separated documents mentioned above. The teams required to present their software design before the implantation phase to gain additional feedback and to practice their oral and communication skills. The presentation will be graded as part of the final project score.

W3. The final project report is usually around 100 pages and every team members must contribute. To make sure that each team member contributes to the writing of the reports, the assignment of writing is as follows:
- For 3-person team: each team must design a main author for the first document, a different member as the main author for the second document, and another different member as the main author for the third document. They will need to rotate ownership of the different documents to compose the final project report. The three persons need to work together for the presentation after the design phase.
- For 4-person team: because of class size, one or two 4-person teams may exist in the class. Again, a different member of the team will be the main author for the three different documents. The final team member will be the main author to put everything together for the final project report. This final team member is also responsible for the working with the other three team members for the presentation after the design phase.

Technology Intensive SLO Assessment:
T1. There are homework and examinations that cover the concepts. Also, the understanding of the concepts and operations will be demonstrated by the students via their group projects (please see T2 below).

T2. The class will consist of several 3 to 4-person teams. Each team needs to produce four tasks using UML and/or a CASE tool. These four tasks are requirements collection and analysis, system specification, software design, and programming implementation. Each team needs to do a presentation after the design stage. The team needs to hand in three separated documents for the four tasks and a final project report.

T3 and T4.
There are homework and examinations that cover cultural, societal, legal, and ethical implications of technology and software development process. Also, students will be evaluated on the group participation of their team project. They practice legal and ethical behaviors in the software development process as a team.
9. **SUGGESTED READINGS, TEXT, OBJECTS OF STUDY:**


10. **BIBLIOGRAPHY OF SUPPORTIVE TEXTS AND OTHER MATERIALS:**


**Other Material for Study:**

http://www.uml.org (The UML website)
http://argouml.tigris.org (ArgoUML website)
http://www.visual-paradigm.com (Visual Paradigm website)
http://live.gnome.org/Dia (Dia website)
http://www.omg.org (Object Management Group website)

Several handouts, case studies, and news articles in software engineering will be used.
11. **PREPARER’S NAME AND DATE**
   Cyril S. Ku, November 15, 2010 (Technology Intensive); September 27, 2011 (Writing Intensive)

12. **ORIGINAL DEPARTMENTAL APPROVAL DATE:**
   November 11, 2010; September 27, 2011

13. **REVISER’S NAME AND DATE:**
   Cyril S. Ku, November 15, 2010; September 27, 2011

14. **DEPARTMENT REVISION APPROVAL DATE:**
   September 27, 2011