I. Course Title: Software Development and Professional Practice I

II. Course Prefix/Number: CSCI 301

III. Credit Hours: 2.0

IV. Prerequisite(s): CSCI 250 minimum grade C AND CSCI 308 minimum grade C

V. Catalog Description:
Combines a range of topics integral to the design, implementation, and testing of medium-scale software systems including fundamental design patterns, software development processes and project management. Students will form teams and create a design for a medium scale software system. In addition to material on software architecture and engineering, this course also includes material on professionalism and ethical responsibilities in software development and design.

VI. Curricular Relationships:
A required course in the computer science major. This course is a prerequisite for CSci302. CSci301 & CSci302 must be taken such that CSci302 is taken immediately after CSci301. That is, If CSci301 is a Spring semester class CSci302 must be taken in the early summer session. And, if CSci301 is taken in the Fall semester CSci302 must be taken in the following Spring semester. This course enhances content knowledge in the ACM curriculum standards SP2, SP3, SP4, SP5, SE1, SE4, SE5, SE6, SE7, SE8, HC1, GV2.

VII. Student Learning Outcomes:
• Students will demonstrate the application proper human-computer interaction techniques.
• Students will demonstrate a knowledge of fundamental software design patterns
• Students will demonstrate knowledge of the software life-cycle and process models including requirements and specification; software design, software validation, software project management, and software evolution.
• Students will be aware of the social implications of computing.
• Students will demonstrate an understanding of professional and ethical responsibilities in software design and development.
• Students will demonstrate an understanding of risk assessment and implications of software complexity.

VIII. Content Outline:
• Foundations of human-computer interaction: Human-centered development and evaluation; human performance models; accommodating human diversity; principles of good design and good designers; engineering trade offs; introduction to usability testing
• Software processes: Software life-cycle and process models; process assessment models; software process metrics; new trends in software development models.
• Software requirements and specifications: Requirements elicitation; requirements analysis modeling techniques; functional and nonfunctional requirements; prototyping; basic concepts of formal specification techniques.
• Software design: Fundamental design concepts and principles; design patterns; software architecture; structured design; object-oriented analysis and design; component-level design; design for reuse.
• Software validation: Validation planning; testing fundamentals, including test plan creation and test case generation; black-box and white-box testing techniques; unit, integration, validation, and system testing; object-oriented testing; inspections.
• Software evolution: Software maintenance; characteristics of maintainable software; re-engineering; legacy systems; software reuse.
• Software project management: Team management; project scheduling; software measurement and estimation techniques; risk analysis; software quality assurance; software configuration management; project management tools; new trends in software project management.
• Social context of computing: Introduction to the social implications of computing; social implications of networked communication; growth of, control of, and access to the Internet; gender-related issues; international issues.
• Methods and tools of analysis: Making and evaluating ethical arguments; identifying and evaluating ethical choices; understanding the social context of design; identifying assumptions and values.
• Professional and ethical responsibilities: Community values and the laws by which we live; the nature of professionalism; various forms of professional credentialing and the advantages and disadvantages; the role of the professional in public policy; maintaining awareness of consequences; ethical dissent and whistle-blowing; codes of ethics, conduct, and practice; dealing with harassment and discrimination; "Acceptable use" policies for computing in the workplace.
• Risks and liabilities of computer-based systems: Historical examples of software risks; implications of software complexity; risk assessment and management.

IX. Course Procedures/Policies/Grading Scale:
• Homework assignments will be given during the semester.
• The major software design project will comprise a majority of the course grade.
• At least one examination will be given during each semester.
• A final examination will be given at the end of each semester.

X. Required/Recommended Readings:
Various materials presented by the instructor.

XI. Issues Unique to Course: This course is offered during the spring semester of odd years immediately followed by the companion course CSCI 302 offered in the early summer session.

XII. Additional Departmental Issues: None