I. Course Title: Software Development and Professional Practice II

II. Course Prefix/Number: CSCI 302

III. Credit Hours: 2.0

IV. Prerequisite(s): CSCI 301 minimum grade C

V. Catalog Description:
The follow up course for CSci 301 in which students will implement, with their team, a software design from CSci 301. Students will experience project management and software development processes directly. The course will focus on programming techniques and the last stages of the software development process, including design, validation, evolution, human computer interaction, using APIs, software tools and software development environments.

VI. Curricular Relationships:
A required course in the computer science major. This course must be taken immediately after taking 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 class enhances content knowledge in the ACM curriculum standards PF5, HC1, HC2, GV1, SE1, SE2, SE3, SE5, SE6, SE7, SE8, SP3, SP4.

VII. Student Learning Outcomes:
- Students will demonstrate the ability to apply event-driven programming techniques.
- Students will demonstrate the application proper human-computer interaction techniques.
- Students will create a software system with a graphical user interface.
- Students will create a software system following the software life-cycle and process models including requirements and specification; software design, software validation, software project management, and software evolution.
- Students will experience participation in a team software development project and examine the implications of various software project management methods.
- Students will examine the impact of software validation as it applies to a software development project.
- Students will experience the implications of professional and ethical responsibilities relative to a team software development project.
- Students will experience the implications of risk assessment and software complexity relative to a medium scale software development project.
VIII. Content Outline:

- Event-driven programming: Event-handling methods; event propagation; exception handling
- 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
- Using APIs: API programming; class browsers and related tools; programming by example; debugging in the API environment; introduction to component-based computing
- Building a simple graphical user interface: Principles of graphical user interfaces; GUI toolkits
- Graphic systems: Raster and vector graphics systems; video display devices; physical and logical input devices; issues facing the developer of graphical systems
- Graphic systems: Raster and vector graphics systems; video display devices; physical and logical input devices; issues facing the developer of graphical systems
- 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,
- Professional and ethical responsibilities: Community values and the laws by which we live; the nature of professionalism; the role and responsibilities of a software development project team member, codes of ethics, conduct, and practice;

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 early summer session of odd years, immediately following the offering of CSCI 301. Students must complete CSCI 301 the semester prior to taking CSCI 302.

XII. Additional Departmental Issues: None.