COURSE NUMBER/TITLE: CS-448 [354-448] SOFTWARE ENGINEERING

CREDITS: 3

COURSE DESCRIPTION: Software development lifecycle, programming languages and environments, program testing, documentation, software management and organization. Class software development project. Prerequisite: CS-244 Data Structures.

TEXTBOOK: Software Engineering, 10th Ed., by Sommerville (adopted Fall 2015)  
The Mythical Man-Month: Essays on Software Engineering, 2nd Ed., by Brooks (adopted Fall 2015)  
Previous:  
Software Engineering, 9th Ed., by Sommerville (adopted S10)  
Software Engineering: A Practitioner’s Approach, 6th Ed., by Pressman (adopted F05)  
Object-Oriented Analysis & Design, 2nd Ed., by Booch (F99 replacing Pohl)  
Software Engineering: A Practitioner’s Approach, 4th Ed., by Pressman (adopted S01)  
Object-Oriented Software Engineering, 1st Ed. by Bruegge (adopted S01)  
Software Engineering with Ada, 2nd Ed., by Booch  
Object-Oriented Programming Using C++ by Pohl (prior to F97)  
Software Engineering with Student Project Guide by Myhätt and (adopted F93 and replaced by Pressman 8/99.)

COURSE OBJECTIVES:  
To complete the course, the student will:  
1. Study the methods and techniques of the software development process for the entire software lifecycle.  
2. Study the effects of the programming language and programming environment on the software development process.  
3. Study the basic concepts of software management and organization.  
4. Practice the techniques of software engineering by participating in a group software development project.

COURSE OUTLINE:  
1. Software Lifecycle Overview  
2. Software Requirements  
   a. Analysis  
   b. Design and specification  
   c. Validation  
3. Software Design  
   a. Design stages  
   b. Design notations  
   c. Validation  
   d. Design reviews  
4. Software Implementation
a. Effects of programming language
b. Programming practices
   1) Methodologies
   2) Programming style
   3) Programming tools and environments
5. Programming Testing and Debugging
   a. Program verification
   b. Code inspections
   c. Theory of testing
   d. Design test cases
   e. Testing and debugging tools
6. Software Documentation and Maintenance
   a. Documentation standards
   b. Documentation tools
   c. Maintenance guidelines
7. User Interfaces
   a. User psychology
   b. Interface design
8. Management of Software Development
   a. Psychology of programmers and programmer groups
   b. Management practices
      1) Programmer productivity
      2) Team organization
      3) Project planning