Course number and name:	**CS 07321: Software Engineering I**
Credits and contact hours:	3 credits. / 3 contact hours
Faculty Coordinator:	Jack Myers
Text book, title, author, and year:	Software Engineering, Ian Sommerville, 2016.

Specific course information

Catalog description: An introduction to the discipline of Software Engineering. Students will explore the major phases of the Software Lifecycle, including analysis, specification, design, implementation, and testing. Techniques for creating documentation and using software development tools will be presented. Students will gain experience in these areas by working in teams to develop a software system. Proficiency in programming is expected of the students entering this course.

Prerequisites: (CS04222 Data Structures and Algorithms or
CS04225 Principles of Data Structures) **and**

(COMP 01112 College Composition II or
HONR 01112 College Composition II Media Literacy or
ENGR 01201 Sophomore Engineering Clinic I) **and**

(CMS 06202 Public Speaking or
ENGR 01202 Sophomore Engineering Clinic II)

Type of Course: ☒ Required ☐ Elective ☐ Selected Elective

Specific goals for the course

1. **Agile/Scrum.** Students have applied the Agile/Scrum methodology to a development project. This includes fully adhering to Scrum values, assuming Scrum roles, participating in Scrum events and producing Scrum artifacts.

    - ABET (d) An ability to function effectively on teams to accomplish a common goal
    - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice

2. **collaborative development**. Students have worked collaboratively in a team to implement and fully test detailed designs and code.

    - ABET (d) An ability to function effectively on teams to accomplish a common goal

3. **project initiation.** Students have demonstrated an understanding of the key elements of project initiation.

    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
    - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice

4. **requirements documentation**. Students have translated end-user requirements into system and software requirements, in a structured software Requirements Document that combines user requirements and functional requirements.

    - ABET (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
    - ABET (f) An ability to communicate effectively with a range of audiences
    - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice

5. **requirements elucidation.** Students have collected and analyzed user requirements using formalisms such as User Stories, UML, and business process modeling.

    - ABET (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
    - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice

6. **system design and implementation.** Students have identified and applied appropriate software architectures and patterns to carry out high level design of a system. Students have critically compared alternative choices and structured this information in an architectural Design Document. Students have implemented an application as specified in the Design Document.

    - ABET (a) An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline
    - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

- ABET (f) An ability to communicate effectively with a range of audiences
- ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice
- ABET (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
- ABET (k) An ability to apply design and development principles in the construction of software systems of varying complexity

7. **technical presentations**. Students have reflected on the outcome of a project and presented their reflections in a verbal report.

   - ABET (f) An ability to communicate effectively with a range of audiences

8. **testing methodology**. Students understand how to produce comprehensive test plans and scripts.

   - ABET (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
   - ABET (f) An ability to communicate effectively with a range of audiences
   - ABET (i) An ability to use current techniques, skills, and tools necessary for computing practice

Required list of topics to be covered

1. Software processes
2. Agile software development and scrum
3. Requirements engineering
4. Design documentation and processes
5. Architecture documentation
6. Software testing and maintenance
7. Utilizing GIT

Optional list of topics that could be covered

8. Formal validation: IQ, OQ and PQ
9. Project management using Microsoft Project
10. UML diagramming tools
11. Creating physical database diagrams
12. RESTful web services
13. ITIL