Dept Number	CS 435	Course Title	Software Engineering
Semester Hours	3	Course Coordinator	Michael Wainer
Catalog Description	Principles, practices and methodology for development of large software systems. Object-oriented principles, design notations, design patterns and coping with changing requirements in the software process. Experiences with modern development tools and methodologies. A team project is an integral part of this course.		

Textbooks

Pressman; Software Engineering : A Practitioner's Approach. McGraw Hill, 7th Edition
ISBN: 9780073375977.

References

Various references to tool and language documentation, resources on patterns, principles, etc.

Course Learning Outcomes

- To understand and develop experience working within a collaborative team environment.
- To become familiar with concepts of software development methodologies and notations.
- To be able to apply modern development tools and practices to create software both individually and collaboratively.
- To understand basic principles of Object Oriented design and the value of software patterns.

Assessment of the Contribution to Program Outcomes

Outcome →	1	2	3	4	5	6	7	8	9	10
Assessed →		X	X	X	X	X		X		

Prerequisites by Topic

CS 330 with a grade of *C* or better; CS 306 with a grade of *C* or better recommended

Major Topics Covered in the Course

1. Introduction to software development {2 classes}
2. Perspectives on software process {3 classes}
3. Introduction to software best practices {3 classes}
4. Communication, collaboration and teamwork {6 classes}
5. Software development tools & environment IDE, testing framework, build scripts {3 classes}
6. Coding style and conventions {2 classes}
7. Object oriented principles {5 classes}
8. Practices and process in depth {6 classes}
9. Design notations {3 classes}
10. Software design patterns {5 classes}
11. Anti-patterns {2 classes}

Major Assignments:

Practice Project to get familiar with basic development practices and tools. (Typically 2 person teams building a game application based upon user stories using JUnit for test driven development.)

Customer Project typically involves merging the entire class into the development team while working with an external customer to develop software using an agile development methodology. Customers may be other research faculty.