CSCE 593 | Software Engineering

Dr. Scott Nykl • Office: Building 642 Room 203
https://git.nykl.net/csce593 • scott.nykl@afit.edu
(937) 255 3636 ext 4395
Fall 2019 • 4 Credit Hours • Grading: Regular (R)

Course Information

Meet Time Tue & Fri, 1000 – 1200

Location Building 646 Room 220

Textbook Design Patterns: Elements of Reusable Object-Oriented Software

Authors: Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
ISBN: 978-0-201-63361-0
Publisher: Addison-Wesley Professional, 1995
Textbook style: Describes commonly used design patterns to solve recurring problems in software development.

Course Description This course is concerned with the development of large-scale software systems (i.e., systems which have a large software component). Techniques in software requirements elicitation, object-oriented design, implementation, quality assurance, and project management are presented, along with discussion of the software development process. Emphasis is on object-oriented modeling using a subset of the Unified Modeling Language (UML). Techniques to facilitate the engineering of secure software systems are introduced. This course will enable you to more effectively communicate with software users and developers and make sound management decisions. Hands-on experience is provided through papers, labs, individual homework problems, and a group project.

Grading Scheme Grading is comprised of 3 categories:

- (*50%*) Group Project: A group project emphasizing software engineering principles.
- (*40%*) Individual Assignments: Written assignments that reinforce the design patterns, data structures and algorithms learned in class.
- (*10%*) Final Exam: The final will include all material throughout the course.

Grading		
	B+ [87.0%, 90.0%)	C+ [77.0%, 80.0%)
A [93.0%, 100.0%]	B [83.0%, 87.0%)	C [73.0%, 77.0%)
A- [90.0%, 93.0%)	B- [80.0%, 83.0%)	C- [70.0%, 73.0%)
		F $< 70.0\%$

Course Policies

1. Attendance: Attendance at all class sessions and exams is mandatory for military and civilians assigned to AFIT as full-time students except for extenuating circumstances. Scheduled classes and exams are defined by the instructor and they are documented in the course schedule. Part-time students are expected to attend scheduled classes, and absences should be explained to the instructor. The student should provide advance notice, if possible. (References: Student Handbook, Graduate School Catalog)
2. Academic Integrity: All students must adhere to the highest standards of academic integrity. Students are prohibited from engaging in plagiarism, cheating, misrepresentation, or any other act constituting a lack of academic integrity. Failure on the part of any individual to practice academic integrity is not condoned and will not be tolerated. Individuals who violate this policy are subject to adverse administrative action including disenrollment from school and disciplinary action. Individuals subject to the Uniform Code of Military Justice may be prosecuted under it. Violations by government civilian employees may result in administrative disciplinary action without regard to otherwise applicable criminal or civil sanctions for violations of related laws. (References: Student Handbook, ENOI 36-107, Academic Integrity)
3. Academic Grievance: AFIT and the Graduate School of Engineering and Management affirm the right of each student to resolve grievances with the Institution. Students are guaranteed the right of fair hearing and appeal in all matters of judgment of academic performance. Procedures are detailed in ENOI 36-138, Student Academic Performance Appeals.
4. Testing Policy: There will be one final exam held on the last day of class. Make up exams will only be considered for extenuating circumstances. For example, sleeping through a final will result in a zero. The make up exam will be conducted as soon as possible and held in my office.
5. Late Assignments and Make-Ups: If not already discussed with the instructor, a late assignment will lose 50% of its value per day past due. Beyond one day (24 hours), the assignment can receive at most 0%.
6. Conduct: The policies outlined in the EN Operating Instructions apply to this course. Specific policies include:
   - All work must be accomplished by the student submitting the assignment for a grade
   - Other student's assignments from this or prior quarters *may not be used at any time*
   - If you do use additional material (such as a web resource) that helps you substantially when completing an assignment, *include a statement describing the resource when you turn in the assignment*
   - Exams are not to be released or discussed

CLASS SCHEDULE				
Week	**Day**	**Meeting Topic**	**Pre-Class Chapters to Read**	**Assign Due**
1.a	1 Oct	○ Course Overview ○ Setup GIT (`git>Lectures/00-Setup Git.pptx`) ○ Lec: What is Software Engineering ○ Brief Overview of Group Projects ○ C++ & Git (`http://learngitbranching.js.org/`)	Ch 6	
1.b	4 Oct	○ Lec: Software Life Cycle Models ○ Large Software Systems, Dependencies, & Build Systems ○ Assignment: Use GIT to `Clone` AFTRBurner Engine & Usr into proper directory structure.	Ch 1	
2.a	8 Oct	○ Lec: Static Modeling with UML ○ Assignment: `HW1` Use default module, load WO, set position via key press. ○ GroupProj: `Phase1` Static UML modeling for DefenseDaemon (Radar, Defense Turret, Buildings, Terrain)		`Clone`
2.b	11 Oct	○ Lec: Object Oriented Design & Decomposition (Part 1/2)	Ch 3	
3.a	15 Oct	○ Lec: Object Oriented Design & Decomposition (Part 2/2) ○ Strategy Pattern (Pg 315) ○ Assignment: `HW2` On key press, launch parabolic trajectory from current location to target coordinates. Use `WO`, `MGLIndexedGeometry`, & `IndexedGeometryRibbon`	Ch 4	`HW1`
3.b	18 Oct	○ Observer Pattern (Pg 219) ○ C++ Refresher: Pointers, Memory address ○ `Phase1`: Group design presentations	Ch 5	`Phase1`
4.a	22 Oct	○ Lec: Stateless Dynamic Modeling ○ Lec: Finite State Machines & Dynamic Modeling ○ State Pattern ○ C++ `Lambdas`, `std::function` (support for Strategy & Observer Pattern)	Ch 6	
4.b	25 Oct	○ Lec: Requirements & Use Case Modeling ○ Fast Functionality Demo `HW2` ○ Template Method ○ GroupProj: `Phase2` Radar Detection for DefenseDaemon	Ch 7	`HW2`
5.a	29 Oct	○ Lec: Software Quality Attributes	Ch 8	
5.b	1 Nov	○ Wingman Day - NO CLASS	Ch 3	
6.a	5 Nov	○ Design Patterns: Creational Patterns (Part 1/2) - Abstract Factory - Factory Method (Virtual Constructor)	Ch 3	

6.b	8 Nov	○ Design Patterns: Creational Patterns (Part 2/2) - Singleton - Prototype ○ `Phase 2` Radar Detection Presentation ○ GroupProj: `Phase 3` - Trajectory Tracking (Plotting) - Trajectory Prediction (Plotting) - Smart Ribbon / Radar Radiation Detection	Ch 4	`Phase 2`
7.a	12 Nov	○ Design Patterns: Structural Patterns (Part 1/2) - Adapter - Composite	Ch 4	
7.b	15 Nov	○ Design Patterns: Structural Patterns (Part 2/2) - Decorator ○ Release `HW 3`: Design Pattern Challenge	Ch 5	
8.a	19 Nov	○ Design Patterns: Behavioral Patterns (Part 1/4) - Chain of Responsibility - Command ○ Release `Phase 4`: Complete Missile Defense System. See Phase 4 Guidance.		`Phase 3`
8.b	22 Nov	○ Design Patterns: Behavioral Patterns (Part 2/4) - Memento - Mediator		
9.a	26 Nov	○ Team Meetings for Project Development	Ch 5	
9.b	29 Nov	○ Thanksgiving - NO CLASS		
10.a	3 Dec	○ Design Patterns: Behavioral Patterns (Part 3/4) - Iterator - Generic Programming		
10.b	6 Dec	○ Design Patterns: Behavioral Patterns (Part 4/4) - Generic Programming ○ Antipatterns of Design Patterns		`HW 3`
11	13 Dec	○ Final Exam Slot: 0730 - 1030		`Phase 4`