

CSCE 247 - Fall 20XX

Software Engineering

Instructor:
Office Hours:
Lecture:

Bulletin Description

Fundamentals of software design and development; software implementation strategies; object-oriented design techniques; functional design techniques; design patterns; design process; source control; testing.

Prerequisite: C or better in CSCE 146

Course Description

Software engineering is concerned with the development and evolution of high-quality software systems in a systematic, controlled, and efficient manner. Software engineers are concerned with safety and reliability of the product as well as the cost and schedule of the development process. The lectures and the group projects will cover all aspects of the software life cycle, from development team management, problem specification and analysis, system design techniques, implementation and documentation practices, testing, to maintenance and evaluation of the final product.

Required Readings

Primary:

- *Software Engineering*. Ian Sommerville. Tenth Edition (Ninth Edition is acceptable).
 - An electronic version of this textbook is available from <http://www.mypearsonstore.com/bookstore/software-engineering-9780133943030>
 - Note: I do not assign homework from the book. It is the primary basis for lectures, and is used as additional reading and back-up. Therefore, you do not have to purchase a copy. I recommend picking it up, but it is optional.

Secondary

- *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Martin Fowler. Third Edition.
- *Head First Design Patterns*. Eric Freeman, Bert Bates, Kathy Sierra, Elisabeth Robson.

Both books are purely optional, and serve as reference books for course content. Both may be useful in this class, and in future software projects.

Additional readings will be assigned as the course progresses. These readings will be available on the course web page or handed out in class.

Learning Outcomes

1. The students will be able to distinguish between software development processes and choose an appropriate process for a particular project.
2. The students will know how to use source control and process management tools.
3. The students will be familiar with requirements elicitation and be able to create a requirements specification document.
4. The students will learn about software architectural models and be able to analyze how control and data flow through a system.
5. The students will understand the principles of object-oriented software design, including how to describe and model the structure of a system.
6. The students will understand and be able to apply software design patterns.
7. The students will be familiar with the fundamentals of requirements-based and structure-based software testing and the accompanying test selection methods.

Exams

There will be an in-class midterm examination, as well as a final exam. All exams are closed-book. APOGEE students can take the exam in-person or online.

- Midterm: <Date>
- Final: <Date>

Assignments

We will assign several homework assignments throughout the semester. Individual assignments will be due approximately every other week. Almost all assignments will include a programming component. Students, working in small groups, will also complete a larger-scale semester-long project.

You need to pull your weight on all group assignments. Substandard work is obvious to your fellow students and the instructor and will be reflected in your grade. Peer evaluations will be turned in throughout the semester, and will be used during grading. Additionally, the exams will reflect project content. If you have not done the project work, you *will* do poorly on the exams.

Grading

40% of your grade will be based on the individual assignments and group project. You are graded on the quality of the work you produce, not on how many hours a week you spend. The details of how much each deliverable is worth will be announced with the projects and assignments. The midterm will be worth 20%, and the final will be worth 30%. The remaining 10% will be awarded based on in-class activities and participation in the group.

Students are required to perform satisfactorily on both exams and assignments to receive a passing grade. All assignments and tests will be awarded 100 points. A general guideline for grading will be the following:

100 >= score >= 90 A
90 > score >= 87 B+
87 > score >= 80 B
80 > score >= 77 C+
77 > score >= 70 C
70 > score >= 67 D+
67 > score >= 60 D
60 > score >= 0 F

Course Outline

Approx. 28 classes total.

Introduction (1 class)

- Fundamental topics

Development Processes: Waterfall, Agile, others. Comparison (2 classes)

- Risk Management
- Waterfall
- Agile (evolutionary, iterative, scrum, XP, etc.)

Source Control, Process Management Tools (1 class)

- Centralized vs Distributed
- Git internals (or similar)
- Bug-trackers
- Issues, Pull Requests, Code Reviews, as an example of how we use tools to implement these processes

Requirements Specification (3 classes)

- Writing a good Requirements document
- Requirements elicitation
- Use-Cases and User Stories

Architecture Design Fundamentals (1-2 classes)

- Layered model
- Repository model
- Client-server model
- Pipe and filter
- Event-driven model
- Interrupt-driven model

Object-Oriented Design (4 classes)

- Object Modeling: how to go from problem description to OO Java code.
- UML class and sequence diagrams

Design Patterns (4 classes)

- Observer, Visitor, Factory, Singleton, MVC, etc.
- Implement Design Patterns in Java.
- Show how popular platforms and libraries (Android, Angular, iOS, etc) use patterns.

Functional Programming (2 classes)

- Lambdas, map, reduce, etc: how to go from problem description to functional code. Use Java 8 lambdas.

Testing Fundamentals (6-7 classes)

- Testing Stages (unit, integration, system)
- Functional Testing
- Structural Testing
- Writing unit tests, Test automation
- Reliability/When to stop testing

Program Maintenance (1-2 classes)

- Refactoring
- Code Smells

Policies and Procedures

This section contains some general rules that will be enforced during this course. Please review these guidelines carefully. The course is governed by the policies in the [Carolina Community: Student Handbook & Policy Guide \(http://www.sa.sc.edu/carolinacommunity/judicial/\)](http://www.sa.sc.edu/carolinacommunity/judicial/). Violations of this code can result in actions varying from a failing grade to expulsion from the university.

Integrity and Ethics

The homework and programs you submit for this class must be entirely your own. If this policy is not absolutely clear, then please contact me. Any other collaboration of any type on any assignment is not permitted. It is also your responsibility to protect your work from unauthorized access. Any violation of this policy will result - at minimum - in a 0 on the assignment. Further infractions will result in a failing grade in the course and further disciplinary action.

Remember that the first tenet of the Carolinian Creed is, "I will practice personal and academic integrity."

Classroom Climate

All students are expected to behave as scholars at a leading institute of technology. This includes arriving on time, not talking during lecture (unless addressing the instructor), and not leaving the classroom before the end of lecture. Disruptive students will be warned and potentially dismissed from the classroom.

Make-Up

The midterm (held during regular lecture hours) and final are required. If any of the tests fall on a religious holiday, the tests will be rescheduled.

Make-ups for graded activities may be arranged if your absence is caused by a documented illness or personal emergency. A written explanation (including supporting documentation) must be submitted to your instructor. If the explanation is acceptable, an alternative will be arranged.

Whenever possible, make-up arrangements must be completed prior to the scheduled activity. A student not taking an exam or not turning in an assignment will receive a score of 0. Alternative times for the final exam will be arranged only under university criteria for rescheduling a final exam.

Late Submissions

Homework assignments are due at the time noted on the assignment handout. Late work is not accepted without prior approval. Any assignment turned in after the due date will be considered late and will be subject to a penalty of 10% per day, including weekends and holidays. Submitting all assignments is a necessary condition for passing this class.

Attendance Policy

Attendance will not be formally checked. However, be aware that the course does have a participation grade. Failing to take part in too many of the in-class activities may result in loss of participation credit.

Special Needs

It is university policy to provide, on a flexible and individual basis, reasonable accommodations to students that have disabilities that may affect their ability to participate in course activities or to meet course requirements. If you have a disability and may need accommodations to fully participate in this class, contact the Office of Student Disability Services: 777-6142, TDD 777-6744, email sasds@mailbox.sc.edu, or stop by LeConte College Room 112A. All accommodations must be approved through the Office of Student Disability Services.

Diversity

Someday you will graduate, and in the real world, you will have to work with a wide variety of people. Now is the time to abandon preconceived prejudices about others. Students in this class are expected to respectfully work with all other students, regardless of gender, race, sexuality, religion, or any other protected criteria. There is a zero-tolerance policy for any student that discriminates against other students.

Computer Science

Relation of Course Outcomes to CAC Student Outcomes*