

CSCE 742 - Syllabus

1: Course Overview

Course Name: CSCE 742 - Software Architectures

Semester: Fall 2018

Instructor: Greg Gay (ggay@cse.sc.edu)

Lecture Hours: Tuesday/Thursday, 8:30-9:45 AM, 2A05 Swearingen Engineering Center

Office Hours: Tu/Th, 4:00-5:00 PM, 2247 Storey Engineering and Innovation Center

Website: <https://dropbox.cse.sc.edu/course/view.php?id=95> (main page)
<http://greggay.com/courses/fall18csce742/> (static backup, usually out of date)

Course Description

Our society is built on software. It powers our homes, it manages our private information, it controls our cars, it automates our factories, and it even regulates our bodies. It is incredibly important that we construct robust, operational systems, especially given growing demand for features, limited development budgets and strict time constraints.

In the beginning, the entire codebase for a system would be written in a single file (or a punchcard!). Then, languages such as C added the ability to link code from multiple files together, enabling easier organization and code reuse. Object-oriented languages further increased our ability to organize code into logical entities. The systems that power our modern society may be built from hundreds - if not thousands - of individual classes.

Software architecture is one of the most important activities performed when designing a system. It is the practice of partitioning a large system into smaller ones that can be created separately, that individually have business value, and that can be straightforwardly integrated with one another and with existing systems. The goal of this course is to master skills that support this partitioning.

We will discuss the purpose and role of architecture in the overall process of software development, both as a process (*architecting* a system) and an artifact (the architecture of a system). We will also examine and debate the similarities and differences between "design" and "architecture". We will examine notations and tools designed to assist software architects and processes that can lead to good architectural outcomes and architectural refactoring.

Outline of Topics to be Covered

- Introduction and Fundamentals of Architecture (1 week)
- Introduction to Viewpoints, Perspectives, and Architectural Definition (1 week)
- Context, Concerns, Stakeholders, and Quality Attributes (1 week)

- Scenarios and Components (1 week)
- Architectural Styles (4 weeks)
- Viewpoints (2 weeks)
- Perspectives (2 weeks)
- Other Topics (1-2 weeks)

See course schedule for specifics.

Learning Outcomes

1. The students will be able to work from stakeholder requirements to create system interfaces that support partitioning.
2. The students will be able to use different viewpoints to document software architectures to different stakeholders.
3. The students will be able to understand architectural quality attributes and how to use perspectives to assess how well the architecture meets them.
4. The students will be able to apply and understand architectural patterns to quickly examine architectural alternatives and choose between them.
5. The students will be able to clearly present and advocate architectural ideas.

Textbooks

This course has no required textbook. However, the following books are the source of most of the covered material and will provide background not covered in class.

- *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives* (Second Edition). Nick Rozanski and Eoin Woods. Addison Wesley, 2012. ISBN 978-0321718334.
 - https://amazon.com/Software-Systems-Architecture-Stakeholders-Perspectives/dp/032171833X/ref=sr_1_2?ie=UTF8&qid=1533577641&sr=8-2&keywords=software+systems+architecture
- *Software Architecture in Practice* (Third Edition). Len Bass, Paul Clements, Rick Kazman. Addison Wesley, 2013. ISBN 978-0321815736.
 - https://amazon.com/Software-Architecture-Practice-3rd-Engineering/dp/0321815734/ref=pd_sim_14_1?encoding=UTF8&pd_rd_i=0321815734&pd_rd_r=37GFCR3DNWZ85KJ38WAJ&pd_rd_w=xUEit&pd_rd_wg=luXhO&psc=1&refRID=37GFCR3DNWZ85KJ38WAJ

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

2: Course Requirements and Grading

Requirements

CSCE 740 is a required prerequisite for this course.

You should have knowledge of some higher level programming language (such as C, Java, and JavaScript). Programming assignments and code examples for the course will primarily use Java; therefore, you should be able to pick the language up on your own if you do not already have experience in it. You are expected to understand basic data structures (such as lists, sets, and trees), algorithms (such as sorting, searching, and tree traversals), recursion, data abstraction, and finite state machine models. Knowledge of UML class diagrams is also assumed.

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: Thursday, October 11, in class (subject to change)
- Final: Tuesday, December 11, 9:00 - 11:30 AM

Homework Assignments and Group Participation

We will assign several homework assignments throughout the semester. These assignments will be completed in groups of three students. APOGEE students may choose to work on the group projects alone, but are encouraged to form groups with in-class and other APOGEE students on the forum on the course webpage. There will be additional reading and quiz assignments to be completed individually.

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.

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.

Grading

Your grade for the course will be calculated from the following components:

- Group Assignments **(40%)**
- Individual Reading Assignments **(10%)**

- Midterm Exam (20%)
- Final Exam (20%)
- In-Class and Group Participation (10%)
 - APOGEE students are expected to submit the in-class exercises within seven days of lecture videos being posted, unless prior permission is given.

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

Total Score	Letter Grade
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

3: 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/>). 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.

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.

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

This is a graduate level course, and attendance will not generally be checked. However, be aware that the course does have a participation grade. Failing to take part in the in-class activities may result in loss of participation credit. During the presentations at the end of the semester, in-class students must attend class and attendance will be checked.

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. Students with disabilities are encouraged to contact their instructor early in the semester to discuss their individual needs for accommodations.

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.