



Carnegie Mellon University
Master of
Software Engineering

17-635: Software Architecture

MW 10:40 -12:00pm, SCR 300 S. Craig Street rm 265

Course Zoom Link (first 2 weeks):

<https://cmu.zoom.us/j/91986266543?pwd=TFHRXJDYmUwMGNMVFQycTVGZWl5UT09>

Passcode: 724496

F 10:40 – 12:00pm,

[A3, Spring 2021, 6 Units]

Last Update: 1/28/21

Instructor	Email	Office Location & Hours
Matthew Bass	mbass@andrew.cmu.edu	SCR 267 Zoom ID: 3609899473 By appointment

Teaching Assistants

Ariel Xiao	yuweix@andrew.cmu.edu	TBD
Ami Vipin Kotia	akotia@andrew.cmu.edu	TBD

Course Description. Architectures for Software Systems aims to teach you how to design, understand, and evaluate systems at an architectural level of abstraction. By the end of the course, you will be able to:

- Understand the influence of architectural drivers on software structures.
- Understand the technical, organizational, and business role of software architecture.
- Identify key architectural structures (styles, patterns, tactics, etc.).
- Understand the principles of good architectural documentation and presentation.
- Understand the impact that open source and third party components have on architectural designs.
- Generate architectural alternatives in a given context and choose among them.
- Understand how formal notations can be used to specify architectures.
- Evaluate the fitness of an architectural design in meeting a set of system requirements and balancing quality tradeoffs.
- Be aware of the future trends in software architecture.

Prior Knowledge. Students are expected to be familiar with programming in at least one, preferably object-oriented, programming language. Formal programming language training is not required. Students need not have any formal background in algorithms, data structures, analysis, or detailed design techniques and methods.

Previous coursework in computer science (such as data structures or algorithms) is not necessary. However, students should have some experience writing small programs or software applications. Students in doubt regarding their experience should obtain instructor's permission.

Learning Objectives. Architectures for Software Systems aims to teach students how to design, understand, and evaluate systems at an architectural level of abstraction. By the end of the course, students should be able to:

- Understand what drives software architecture, which means:
 - Define the following concepts: constraint, functional requirement, and quality attribute.
 - Be able to determine if a given requirement is a constraint, functional requirement, or quality attribute.
 - Specify an actionable quality attribute requirement according to the 6-part scenario framework for at least the following quality attributes:
 - Availability, Performance, Modifiability, Security, Scalability
 - For functional requirements, students should be able to specify actionable requirements to include:
 - Pre & post conditions, Alternative flows, Error conditions
- Understand the relationship between business goals and architectural drivers, which means:
 - Be able to identify and characterize an architectural driver
 - Be able to determine if goals are articulated in sufficient detail to derive suitable drivers
 - Understand the impact that a failure to achieve a given driver would have on the business goals
- Understand architectural structures, which means
 - Be able to define:
 - Dynamic structures, Static structures, Allocation relationships
 - Be able to identify the software elements in dynamic and static structures
 - Be able to identify the characteristics of each element
 - Be able to identify the structures that are related to the following quality attributes:
 - Modifiability, Availability, Security, Scalability, Performance

- Understand tactics, which means:
 - Be able to identify the tactics for each of the following quality attributes:
 - Modifiability, Availability, Security, Scalability, Performance
 - Be able to apply any of the (appropriate) tactics in a specific context for the following quality attributes:
 - Modifiability, Availability, Security, Scalability, Performance
- Understand what architectural patterns are, which means:
 - Be able to define the term “architectural pattern”
 - Be able to tell the difference between architectural patterns and non-architectural patterns
 - Be able to related architectural patterns to architectural tactics
 - For at least the following architectural patterns, students should be able to identify the elements, topology, semantics, properties promoted and properties inhibited:
 - Pipes and filters, Client server & N-tiered patterns, Dataflow pattern, Shared information systems, Service oriented patterns, Layered systems, Event-based systems
- Be able to analyze a given architecture with respect to:
 - Modifiability, Availability, Security, Scalability, Performance
- Be able to document an architecture according to the “Views and Beyond” approach (LO7), which means:
 - Be able to determine the purpose of the architectural document
 - Identify the views that are required to support the needs
 - Populate the views appropriately to achieve the intended purpose
 - Organize the document such that it supports its intended purpose
 - Review a document with respect to its fitness for purpose
- Understand how to operationalize architectural practices in different lifecycles, which includes:
 - “Project” oriented lifecycles
 - UP-like processes
 - Agile practices
 - ACDM principles
 - Continuous deployment lifecycles

Learning Resources. The course and all course materials will be distributed online and accessible with a CMU account via Canvas.

Assessments. Students learn more by applying and explaining ideas to others, thus, the course requires the following activities:

- **Lecture and Reading Assessments:** These are short online questions derived from the required readings and lectures
- **Class participation:** There are exercises students will do in recitations in pairs or small groups to practice applying the concepts learned in the course as well as participation in online discussions in Piazza
- **Individual Homework Assignments:** These will be primarily programming and reflection assignments based on the concepts learned throughout the course
- **Final Exam**

Assessment	Final Grade %
Quizzes	30%
Individual Homework Assignments	30%
Final Exam	30%
Class participation	10%

Course and Grading Policies

- **Late-work policy:** All work is expected to be handed in at the indicated due date and time. For fairness to the whole class, no late submissions or makeups will be accepted for the Quizzes. We will, however, drop the lowest grade. The penalty for turning in Individual Homeworks late is 10%/day. In the first week of classes, you should receive a course schedule for each course; please use them to plan ahead. If you have any questions you should raise them immediately rather than waiting for conflicts to arise.
- **Participation policy.** Class participation will be graded by in-class engagement, including asking relevant questions based on a critical review of required readings and lectures, preparation for any in-class exercises, and responses on the class discussion board. The lack of attendance and participation, will count against your participation grade.

Course Schedule. The following schedule provides a general overview of topics and assignments. Please refer to the syllabus online in Canvas for specific lecture topics, reading assignments and due dates.

Date	Topic	Assignments Given	Assignments Due	Readings Due
2/1	Introduction (remote)			
2/3	Architecture Uses (remote)	A1		
2/5	Recitation (remote)		Q1	
2/8	Architectural Requirements (remote)			
2/10	Architectural Structures (remote)	A2	A1	
2/12	Recitation (remote)		Q2	
2/15	Architectural Patterns			
2/17	Modifiability	A3	A2	
2/19	Recitation		Q3	
2/22	Performance			
2/24	Availability	A4	A3	
2/26	Recitation		Q4	
3/1	Attribute Driven Design I			
3/3	Attribute Driven Design II	A5	A4	
3/5	Recitation		Q5	
3/8	Architecture Centric Practices			
3/10	Documentation	A6	A5	
3/12	Recitation		Q6	
3/15	Architecture Reuse/Frameworks			
3/17	Architecting Intelligent Systems		A6	
3/19	Final Exam			

Accommodations for Students Disabilities. If you have a disability and have an accommodations letter form the Disability Resources office, I encourage you to discuss your accommodations and needs with me as early in the semester as possible. I will work with you to

ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, I encourage you to contact them at access@andrew.cmu.edu.

Academic Integrity. Honesty and transparency are important to good scholarship. Plagiarism and cheating, however, are serious academic offenses with serious consequences. If you are discovered engaging in either behavior in this course, you will earn a failing grade on the assignment in question, and further disciplinary action may be taken.

For a clear description of what counts as plagiarism, cheating, and/or the use of unauthorized sources, please see the [University's Policy on Academic Integrity](#).

If you have any questions regarding plagiarism or cheating, please ask me as soon as possible to avoid any misunderstandings. For more information about Carnegie Mellon's standards with respect to academic integrity, you can also check out the [Office of Community Standards & Integrity](#) website.

Student Wellness. As a student, you may experience a range of challenges that can interfere with learning, such as strained relationships, increased anxiety, substance use, feeling down, difficulty concentrating and/or lack of motivation. These mental health concerns or stressful events may diminish your academic performance and/or reduce your ability to participate in daily activities. CMU services are available, and treatment does work. You can learn more about confidential mental health services available on campus at the [Counseling and Psychological Services](#) website. Support is always available (24/7) from Counseling and Psychological Services: 412-268-2922.

We must treat every individual with respect. We are diverse in many ways, and this diversity is fundamental to building and maintaining an equitable and inclusive campus community. Diversity can refer to multiple ways that we identify ourselves, including but not limited to race, color, national origin, language, sex, disability, age, sexual orientation, gender identity, religion, creed, ancestry, belief, veteran status, or genetic information. Each of these diverse identities, along with many others not mentioned here, shape the perspectives our students, faculty, and staff bring to our campus. We, at CMU, will work to promote diversity, equity and inclusion not only because diversity fuels excellence and innovation, but because we want to pursue justice. We acknowledge our imperfections while we also fully commit to the work, inside and outside of our classrooms, of building and sustaining a campus community that increasingly embraces these core values.

Each of us is responsible for creating a safer, more inclusive environment.

Unfortunately, incidents of bias or discrimination do occur, whether intentional or unintentional. They contribute to creating an unwelcoming environment for individuals and groups at the university. Therefore, the university encourages anyone who experiences or observes unfair or hostile treatment on the basis of identity to speak out for justice and support, within the moment of the incident or after the incident has passed. Anyone can share these experiences using the following resources:

- **Center for Student Diversity and Inclusion:** csdi@andrew.cmu.edu, (412) 268-2150
- **Report-It** online anonymous reporting platform: reportit.net username: *tartans*
password: *plaid*

All reports will be documented and deliberated to determine if there should be any following actions. Regardless of incident type, the university will use all shared experiences to transform our campus climate to be more equitable and just.