Spring 2024

CS 685-102: Software Architecture

Prabhat Vaish

New Jersey Institute of Technology University Heights Newark, NJ 07102
http://njit.edu

CS 685 Software Architecture – Spring 2024

Part I: Course and Instructor Information

Semester	Spring 2024
Course	CS 685
Instructor	Prabhat Vaish
Course Meeting	TBD
Office Hours	https://njit.webex.com/meet/pvaish (Links to an external site.) Available Tue between 4-6 PM

Part II: Course Description

1. Course description:

The software architecture defines the structure and interactions of software modules. This course provides a working knowledge of the terms, principles, and methods of software architecture and module design. It explains the constraints on the design and the properties of capacity, response time, and consistency. The "4+1" architecture model is taught with architectural styles, interface isolation, decoupling, reuse, agile design with software patterns, data structures, queuing effects, design simplification, and refactoring. The non-functional requirements of reliability, performance and power consumption, component-based design and good industry practices for documenting and managing the architectural process are taught.

The course on software architecture deals with the concepts and best practices of software architecture. The focus is on theories explaining the structure of software systems and how the system's elements are meant to interact given the imposed quality requirements.

Topics of the course are:

- Architecture influence cycles and contexts
- Technical relations, development life cycle, business profile, and the architect's professional practices
- Quality attributes: availability, modifiability, performance, security, usability, testability, and interoperability
- Architecturally significant requirements, and how to determine them
- Architectural patterns in relation to architectural tactics
- Architecture in the life cycle, including generate-and-test as a design philosophy, architecture conformance during implementation
- Architecture and current technologies, such as the cloud, social networks, and mobile devices
- Architecture competence: what this means both for individuals and organizations

For the latest course information go to CS 685 102 - Software Architecture
The information below should help you plan and organize your preparation during the semester.

2. Prerequisite courses and knowledge:

- Prerequisite course: None
- Required background:
  To be successful in this course, you should either have successfully taken an undergraduate software engineering course, or CS 505/506. Alternatively, an experience in developing industrial strength software will be good help.
  - The students are required to have knowledge of key systems concept, software development life cycle, and programming in Java or Python or a similar language.
  - Good understanding of programming, design, development, data modeling techniques and database fundamentals is expected as well.
  - Good understanding of modern trends in information analysis, information technology, cloud computing, object-oriented principles and agility are a plus
  - Undergraduate software development courses provide a good foundation

3. Outcomes expected upon the completion of the course:

- Know the key components and structure of software architecture
- Create and document multiple views of a software architecture
- Use architecture to manage and control software attributes
- Be familiar with and know how to apply architecture from industry
- Analyze and develop the appropriate architectural approach and structure for various technical, business and product situations
- Know how to design, implement and manage the software architecture function in an organization

4. Assessment throughout the course:

Assignments consist of Assigned Readings, problem sets, class presentations, and working as part of a team completing an architecture project (Projects). Each of the six teams will work on a project related to analysis & design or, green-field development or, maintenance or, a critique of an existing product, service or framework. Each team will be making three presentations (analysis, design alternatives and, final report) to the class on their respective projects. For twelve weeks, starting from week 3, three students will present on a pre-assigned topic to the class.

- Term project execution and deliverables - content, mastery of methods discussed in class and creativity; teamwork; research and analysis skills
- Quiz and assignments - content, understanding of methods discussed in class and their effective user or application to the assignment, research and analysis skills
- Class participation – open contribution to the discussions and exercises, sharing, collaboration

5. Required & Recommended texts:

- ***Lecture Notes***
  Lecture notes are the basic course material for this class. The notes are made available on Canvas every week.
- ***Textbook***
  There are no textbooks for this course.
- ***Articles and Discussion Supporting Materials***
  For the list of readings check the Course Outline available on Canvas as well as on Discussions forum.

6. Other Web resources:

See Class information on Canvas (CS 685 102 - Software Architecture)

Part III: Mapping Learning Outcomes to Course Assessment

Course Learning Outcome	Measure (assignment, quiz or project)
Know the key components and structure of software architecture	In class and online discussions; term project
Create and document multiple views of a software architecture	Assignments, Quiz; term project
Use architecture to manage and control software attributes	In class and online discussions; assignments, term project
Be familiar with and know how to apply architecture from industry	In class and online discussions; term project
Analyze and develop the appropriate architectural approach and structure for various technical, business and product situations	In class and online discussions; term project, final exam
Know how to design, implement and manage the software architecture function in an organization	

Grading Procedures

1. Each student will work on a team project to gain hands-on experience about the analysis, design and documentation of a software architecture. Each team will be required to deliver a set of artifacts associated with the product (e.g., project process and associated documents, and other artifacts). The individual student grade on the project will be a combination of an evaluation of the overall completeness and quality of the product deliverables, project presentations and reviews, and the contribution of the individual to the team effort.
2. Individual and team in-class exercises, covering recent assignments and classroom work, will be given on a weekly basis.
3. Individual and team homework will be assigned as necessary.
4. Participation in-class discussion and attending classes is a must in this course
5. Anyone found cheating on an exam, assignment or project will receive an automatic F in the course.
6. There are no make-ups on in-class exercises. Absence from an exam/quiz is excused only in a medical emergency.

NOTE: The expectation is that you will work approximately 12 hours per week on this course; at least 5 of these hours should be on the project. Given that the project lasts 12 weeks, each team member is expected to work on the project *at least* 60 hours. You should be able to accomplish something pretty great in this time; please make the most of this opportunity.

Part IV: Course Outline (Note: this course outline is preliminary and subject to change)

Week	**Lecture/Activity/Discussion**	**Reading (preliminary)**. *Check Canvas for additional reading in every module.*
Week 1	Course logistics and introduction Course introduction – topics, objectives, Teamwork, Communication, Project	Class presentation: Best Practices (discussion on Canvas) Every student is expected to find a paper, survey or a topic discussing one or several current best practices and to provide an outline and references on Canvas. All students are expected to comment on at least 2 postings by other students.
Week 2	Software Architecture – An introduction	Project presentation guidelines and requirements will be provided on Canvas 1) All groups finalized 2) Teams work together to select topic and identify project's key contributions
Week 3	Software Architecture – Key principles	Project start: 3) Project proposal posted on Canvas
Week 4	The Architecture Definition process	
Week 5	Quality Attributes - 1	
Week 6	Quality Attributes - 2	
Week 7	Architectural Tactics and Patterns - 1	
Week 8	Architectural Tactics and Patterns - 2	
Week 9	Project discussions	
Week 10	Architectural Tactics and Patterns - 3	
Week 11	Risk Driven architectures	
Week 12	Documenting Software architecture	
Week 13	Architecture Evaluations	
Week 14	Architecture in special circumstancs	Special focus on blockchain and LLM architectures.
Week 15	Final Présentations	

Part V: Assignment Weighting (How Your Final Grade is Being Calculated?)

Assessment Item	Percentage of final grade
Term Project	40%
Quiz & Other assignments	30%
Class participation, Discussions etc.	30%

Part VI: Delivery Mechanism

The following delivery mechanisms will be utilized:

Face-to-face lectures (delivered via Webex)
Canvas: CS 685 102 - Software Architecture

1. Lectures / Class participation - (30%)
   Lectures will be used to introduce and demonstrate topics, providing a basis for assignments. Lectures will be conversational and interactive, and I expect you to participate ask and answer questions, share your opinion, and otherwise **"be all in"** with the course. Note that most lectures have ASSIGNED READINGS which you should do **before** lecture. These readings will provide a theoretical background for the practical topics we'll discuss (and will be the basis of *Quizzes* described below).
   Lecture recordings and supplemental references will be posted in Canvas weekly.
2. Quizzes/HomeWorks - (30%)
   There will be several graded quizzes during the semester. Each quiz is meant to review and test your knowledge of the material covered over several weeks. There is a time limit for each quiz administration.
   HomeWorks will be primarily written assignments, with a couple of programming assignments mixed in. These are intended to give you a chance to practically utilize and experiment with different architectural concepts.
3. Group Project - (40%)
   One of the key learning for students in this class is the feel of real-world teamwork and presentation experience -- The group project is extremely important for your learning in this class. Further details will be provided in Canvas and class.

Group Work:

Group work is an important part of this course. Students will be allowed to self-select into groups (teams) of 4-5 individuals by a certain date. Students not having formed a group will be formed into separate groups (teams) by the instructor. It is expected that all students will contribute equally to the work of a group. Each group submission will include a cover page affirming such.

Recognizing that the semester long group project requires a sustained commitment of all members. If a group finds that a member is not carrying his/her assigned group responsibilities, the group will be able to petition the instructor to have that individual removed. Before this petition is accepted, the group and instructor will meet to discuss. If an individual is removed from a group, he/she will have to complete project steps individually. Please take group responsibilities seriously.

Unexcused late assignment submissions may not be accepted or accepted with penalty.

Part VII: Plagiarism and Academic Integrity

The approved "University Code on Academic Integrity" is currently in effect for all courses. Should a student fail a course due to a violation of academic integrity, they will be assigned the grade of "XF" rather than the "F" and this designation will remain permanently on their transcript.

All students are encouraged to look over the University Code on Academic Integrity and understand this document. Students are expected to uphold the integrity of this institution by reporting any violation of academic integrity. The identity of the student filing the report will be kept anonymous.

NJIT will continue to educate top tier students that are academically sound and are self-disciplined to uphold expected standards of professional integrity. **Academic dishonesty will not be tolerated at this institution.** Potential offences in papers and assignments include, but are not limited to:

- Using someone else's ideas or words without appropriate acknowledgement.
- Submitting your own work in more than one course without the permission of the instructor.
- Making up sources or facts.
- Obtaining or providing unauthorized assistance on any assignment.

Turnitin.com will be used to assist in the evaluation of the originality of some of the term work. Turnitin.com is only a tool which will assist in detecting plagiarism.

Normally, students will be required to submit their course essays to Turnitin.com for a review of textual similarity and detection of possible plagiarism. In doing so, students will allow their essays to be included as source documents in the Turnitin.com reference database, where they will be used solely for the purpose of detecting plagiarism. The terms that apply to the University's use of the Turnitin.com service is described on the Turnitin.com web site - http://turnitin.com/.

Part VIII: Getting Help - General

The IST Helpdesk is the central hub for all information related to computing technologies at NJIT. This includes being the first point of contact for those with computing questions or problems.

There are three ways to contact the Helpdesk:
1. Call 973-596-2900. Monday - Friday 8 am - 7 pm.
2. Go to Student Mall Room 48. Monday - Friday 8 am - 7 pm
3. Log a Help Desk Service Request online - https://ist.njit.edu/support/contactus.php.

Part IX: Getting Help – Canvas, WebEx etc.

PROTOCOL FOR DISTANCE LEARNING

- All instruction shall be delivered through Canvas. All directions and expectations for students will be posted on individual course Canvas pages. Students are responsible for checking each class on Canvas each day and for being especially attentive to email during this time.
- Attendance, participation, engagement, and understanding will all be monitored through submitted work.
- Work will be self-paced and guided by deadlines as designated. Coursework may be assigned and due in "chunks" so as to allow students to work through the material at their own pace while at home.
- All coursework will be submitted to Canvas.

Web-ex participant etiquette

The following participant expectations should be the norm. Students who don't follow these guidelines can be removed from the Webex meeting if necessary.

- Be on time
- Mute your microphone if you aren't talking
- Keep your video on throughout the class. In case of bandwidth issue, the instructor will turn off his video first.
- Raise virtual hand, if you want to speak on a point. You may also be asked specifically to comment on a topic.
- There will be online questions and/or surveys during the class that will be available for a limited duration. Please respond to them as and when asked. Some of these questionnaires will count towards the class participation grades.
- Only post chat messages relevant to the lessons

Respect

This class may involve discussion of topics on which you and your classmates may have differences in opinion. Please be respectful of others (students or otherwise) at all times. Be thoughtful and considerate of others' ideas; do not dismiss them out of hand. There is almost always more than one right answer!

Correspondence

We will send out official course announcements and information by email, so you should check your email daily (it may not be hip, but it's accessible and effective!) Note that this email will go to your njit.edu address. If you prefer to read your email on another account, you should set your NJIT account to forward your email to your preferred account.

You are welcome to email me at any time. When emailing, please **make sure to sign your emails!** This will let me know who is writing and will help us to better answer questions.

Class Attendance

Make every effort to attend each class meeting! Class will begin and (usually) end on time. Please do your best to get to class before the start of the session. Students are expected to attend all meetings, with exceptions permitted in case of illness and family emergencies. Please silence all cell phones/pagers/etc. before the beginning of each class.

Collaboration

We encourage collaboration. It is always permissible, even desirable, to talk with your classmates about the course material or the requirements of an assignment. Feel free to discuss your approaches to a problem, or get feedback on design, help with syntax errors, etc.

But "collaboration" does *not* mean copying other people's code and trying to pass it off as your own. Collaboration means talking through your approach to a problem or showing someone how you make something work. You can borrow approaches or techniques, but the final product (the application/project you create and submit) must be your own.

In other words: talk with others, but then take a break before you actually do the work on your own. This process assures that you are able to reconstruct what you learned from the collaboration afterwards all by yourself. The most important part of the assignment is the process of the getting the solution—_including _the false starts, bugs, misconceptions, and mistakes—because the learning occurs in the doing. Completely apart from the ethical issues, copying a solution deprives you of the whole point of the assignment.

We will be looking at and using a lot of open-source code in this course. Although professional developers often reuse code they find on the web, they also take the time to understand what that code is doing, customize it to their specific context, and cite the source so that they can find it again later. If you want to use code you find on the web, you **must** include a reference to where you found the code (a URL in a comment is fine) and take the time to understand why it works. Otherwise, you won't learn anything. For example, there are lots of code skeletons for creating interactive visualizations on the web. It's fine to use these, just be honest about it. Failing to give appropriate credit is a form of plagiarism, and so is considered cheating.

Special Accommodations

I also encourage all students having difficulty, whatever the reason, to consult privately with me at any time.