

cs4471A & cs9549A

Software Design and Architecture

Course Outline: Fall 2021

Logistics and Instruction:

Class Location	online
Day and Classtime	Fr 10.30 AM – 1.30PM
Instructor	Nazim H. Madhavji
TA	Mike Rist
Emails to the instructor	<my last name> <<at>> gmail <<DOT>> com
Emails to the TA	<m> then <last name> <<at>> uwo <<DOT>> ca
Office Hours (instructor):	TH: 1.30 PM –2.30pm Catch me right after the class!

Sessional Dates

Term begin	Wed 8 th September, 2021.
First class	Fri 10 th September, 2021.
Last class	Thu 8 th April, 2021.
Term end	Wed 8 th December, 2021.
Thanksgiving	Mon., 11 th Oct., 2021
Reading week	Mon 1 st Nov. 2021 – Sun 7 th Nov. 2021

Students' comments: [click here.](#)

Noteworthy points: [click here](#)

Quick Access:

[Course Description](#)

[Expectations](#)

[Prerequisites](#)

[Textbooks](#)

[Evaluation](#)

[Topics](#)

[Back to the Top of the document](#)

Course Description

The objective in this course is to share knowledge and technology on “software architectures”.

A software architecture is an abstract representation of a software system, filtering out what is traditionally considered detailed design- and implementation-level issues (such as: algorithm, design patterns, data representation and coding) and highlighting such aspects as the overall system structure, inter-relationships and interaction among these the elements in the system structure, and other “run-time” or “off-line” properties of the system (such as reliability, performance, portability, inter-operability, etc.).

Whereas the *requirements* of a system generally state the functionality and the expected behaviour of the deployed system, the *architecture* of the system describes how the desired functions and qualities are to be (or have been) implemented. An architecture sets bounds for lower-level design of the system and gives a technical context to future enhancements of the system.

For a large and complex (software-intensive) system, its architecture is an essential means for controlling and evolving the system. Among many other uses of the architecture include: development team organisation, overall system test planning, training of new developers, release planning, defect analysis, system component reuse, size/cost/effort estimation, and vision sharing with various stakeholders.

Among the resources required to create a high quality system architecture include: requirements of the system; organisational context; domain and technical knowledge and experience; existing system and its architecture (if any); appropriate stakeholders; and architecting notations, methods, techniques, tools and processes.

Just as there are patterns for the design of a system, there are patterns for the architecture of the system. This helps in the selection of the entire, or parts of an, architecture from a set of choices; increasing reuse, cost reduction, and quality development.

Work Components

In this course, the key work components include:

- Weekly assigned **readings** and creating **individual summaries**.
- Creating a **technical domain model** from knowledge gained in the course.

- **Specialist topic**: literature readings and in-class presentation.

Briefly, each team will identify a focused architectural topic (approved by the instructor), search related literature, and present their topic.

- In-class **group summary creation**, and **Q&A and discussions**.
- **Team project**.

Briefly, the project is about creating microservices, architected based on desired quality attributes, on a Cloud platform (your choice). Projects are carried out in teams that will give progress/final reports, demo, and supporting presentation.

Class Operation – IMPORTANT TO NOTE:

- There will be initial, introductory, set of lectures in the class. The class schedule will indicate the exact dates. Beyond the initial set of lectures, formal lectures will only be given if the instructor sees this as appropriate.
- By default, **all the students** are expected to read the assigned weekly literature (typically specific book chapters) and extract key elements therefrom and fill the given **individual templates**. These readings are mandatory. The completed templates will need to be submitted every week. They will be assessed.
- In the class, **groups will create their group template** based on individual templates. These group templates will be posted “live” to OWL. These will then be used to drive “live” Question-Answer (QA) interactions in the class. These (QA) interactions will be assessed in real-time as “class participation” marks.
- **Active learning**: Please note that if you are expecting to come to the class and be lectured throughout the term then this is NOT the course for you. Please drop this course and choose another one. If not sure, please consult the instructor in person for advice.
 - In this course, you will need to spend time reading and understanding the weekly assigned material before you come to the class.
 - In the class, you will be expected to respond to the questions posed and discuss related material. You will be assessed on the quality of your responses.
 - You are thus expected to actively read the assigned material and discuss among your group, and actively engage in the class QA and discussions.

- Clearly, what you will get out of this class is heavily dependent upon what you will put into it.
- Not following this advice will result in wasting your time and in poor learning experience and outcome.

Equipment needed in the class:

- You will need your **laptop** for discussing the individual templates and creating group templates among the members of your group.

Learning Outcomes:

- Familiarity with the notion of software architectures, their importance, and different types of architectures.
- Understanding of the role architectures play in software-intensive systems and in system development.
- Understanding of, and modelling experience with, system and architecture qualities, tactics and patterns (through a class project).
- Understanding of management and governance issues and the relationship between architectures and business.
- Experience with: reading assigned literature and extracting key elements therefrom; formulating trigger questions; creating own comments concerning the “content”; debating technical issues through discussions.
- Experience with creating a technical domain model from the concepts learnt in the course.
- Experience creating micro-services on a cloud platform.
- Experience with researching and synthesizing architecture-related material on a focused topic.
- Experience with presenting findings and results through presentations.

Expectations

- Some expectations were already described earlier (see the important note on **Class Operation**)
- This is not a course where one writes thousands of lines of code,

characterised typically by intense effort of getting the program to work during the days approaching the project deadline. Rather, it is an intellectual course where system-level decisions are represented in a high-level notation (e.g., UML) that together form a system architecture. These decisions are implemented and demonstrated to be operational.

- Such decisions require circumspection of a wide variety of issues (user needs, domain issues, business goals, technical issues, management, regulatory and legal issues, socio-political issues, and others) that can affect the feasibility and quality of the system.
- It also requires collaboration among the project team members to achieve the project-goals. (Typically, in industry, it requires collaboration with various stakeholders such as user representation, requirements analysts, verification specialists, project managers, product and release managers, designers and integrators, and others.)
- In those attending the course, it requires dedication, self-motivation, teamwork, willingness to learn from diverse sources, and ability to communicate and share with others.
- In-class discussions are “triggers” for awareness of concepts in this subject. Students are expected to learn from identified sources, problem solving and group work. In other words, “active learning” also occurs outside the class time.

Prerequisites

- For undergraduate students: CS3307 -- Object-Oriented Design and Analysis. For graduate students, an undergraduate degree with software engineering course(s).
- Also, please note the following regulation from the university:
Unless you have either the requisites for this course or written special permission from your Dean to enroll in it, you will be removed from this course and it will be deleted from your record. This decision may not be appealed. You will receive no adjustment to your fees in the event that you are dropped from a course for failing to have the necessary prerequisites.

Note: Students who have been admitted to this course without the normal prerequisites may not have been exposed to some of the background material expected for this course; it is the responsibility of these students to gain familiarity with this material on their own. These students are encouraged to speak to the instructor for advice.

For the students who are not familiar with Requirements Engineering, it is

strongly recommend that you read at least the following:

Requirements Engineering, by Gerald Kotonya and Ian Sommerville, Wiley, 1998.

- *Chapters 1, 2, 3, 4, 6, 8*

Textbooks:

Required Text book:

Software Architecture in Practice,

3rd edition

Len Bass, Paul Clements and Rick Kazman

Addison-Wesley, 2012

ISBN-13: 987-0-321-81573-6

ISBN-10: 0-321-81573-4

(On-line book is acceptable)

Supplementary Text books:

DevOps -- A software Architect's perspective

Len Bass, Ingo Weber, and Liming Zhu

Addison Wesley, 2015

<http://ptgmedia.pearsoncmg.com/images/9780134049847/samplepages/9780134049847.pdf>

Designing Software Architectures: A Practical Approach

Humberto Cervantes and Rick Kazman

Publisher: Addison-Wesley, 2016 | **ISBN:** 0134390784

Documenting Software Architectures: Views and Beyond

Second Edition

Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, Judith Stafford

Addison-Wesley Professional, 2011

ISBN-10: 0321552687
ISBN-13: 9780321552686

Software Architecture: Foundations, Theory, and Practice

Richard N. Taylor, Nenad Medvidovic and Eric Dashofy

Wiley, 2009

ISBN-10: 0470167742

Software Engineering, An Object-oriented perspective

(Chapter 5 particularly)

Eric J. Braude

Wiley, 2001

ISBN: 0-471-32208-3

Evaluation

- All material covered in the course (including lectures, discussions, assignments and projects, books and other cited resources) is examinable.
- The teaching staff reserve the right to adjust (lower or raise) a student's marks for the tabulated components below based on their judgment of the student's knowledge and understanding of the subject matter during the term.
- Project logistics:
 - Projects will be carried out in groups.
 - The membership of a group will be self-assigned based on the provided skills background. Any adjustments in team membership will be done only at the beginning of the course – by the individuals concerned in conjunction with the rest of the team members. Once formed, the group membership will not be changeable for the rest of the term.
 - Rules for group behaviour, responsibilities, constraints, consequences, etc., will be presented in the class by the instructor.
 - **EXTREMELY IMPORTANT:** In the event a group member is removed from his or her group for reasons of discontentment, please note that placement of that individual in another group will

***not* be possible. In this case, the student concerned would have no choice but to withdraw from the course. PLEASE note that there is nothing else that can be done within the parameters of operation of this course. The project and other groupwork MUST be done as a group and not individually.**

- The grading criteria, as applied to each evaluation component, will be described with the details of the component.
- **Attendance in class is mandatory.** See the table below for consequences of absenteeism.
- Those who miss the in-class question and answer session, etc., will receive zero marks for this component (exceptions only as per the university policy).
- *There will be no makeup Quiz or Test, except for students requesting a Special Quiz or Test for religious reasons. These students must have notified the course instructor, by email, at least 2 weeks prior to the Quiz or Test.*

If you miss the Quiz or Test for any other reason, follow the procedure for Academic Accommodation for Medical Illness. If accommodation is approved by your Dean's office, the Quiz or Test component will be redistributed to the other evaluation components of the course.

- **IMPORTANT: grading will begin on day one. There will be no make up mark for days missed.**

Component	Max. %	Dates
Summaries of assigned readings (pre-add/drop date – Fall term)	15**	Weekly till 12 th Nov., 2021
**Summaries of assigned readings (Post-add/drop date till end of the term). This subsumes the Pre-add/drop date mark obtained.	20	Weekly after 12 th Nov., 2021 till the end of the term.
Topic presentation.	20	As scheduled.
Questions and Answers (in-class interactions)	10	Throughout the term.
Technical domain model resulting from the knowledge gained in the course.	10	Throughout the term.
System architecting project	40	As scheduled.
Lesson learnt in the class AND shared with the class on the designated cloud storage.	-5% (minus 5%) for each class.	Each class after the preliminary lectures.
Class attendance mandatory [NHM: University policy applies] – see *** below.	-5% (minus 5%) for each class missed	Attendance checked by peers.
Show and Tell (S&T) class presentation (voluntary – only beginning of term volunteers accepted. Sorry late or last minute comers! Admin of S&T gets 3% bonus.)	<u>BONUS</u> Effort: (1%); Ave.: (2%); Enlightening/interesting: (3%)	As scheduled.

Please Note:

- Class attendance is *mandatory* (see table above for penalty). Reason is that in EACH class, there is a live exchange of questions and answers. EVERYBODY has to come prepared, as part of groupwork, for every class and participate.
- Thus conflict with any other course will just not work It will NOT be acceptable that you attend the class for only part of the class-duration (of 3 hours).
- It will NOT be acceptable that you miss the class due to reasons such as job interviews, extra-curricular activities, family trips, events, etc. Please arrange your other events accordingly or be prepared to lose 5% per class missed.
- University policy on absenteeism will prevail.

Topics

The topic list is tentative. The actual topics will be dictated by the dynamics of the class schedule.

Topic*	Required Text book section
Course Overview & Preliminaries	
Introduction to software architectures	Part One
Quality attributes	Part Two
Architecture in the life cycle	Part Three
Architecture and Business	Part Four
Newer Architectures	Part Five

* **Note:** other literature may be added to these topics as deemed appropriate.

Email and OWL Policy

Staff contacting students:

- *We may need to send email messages to the whole class, or to students individually. Email will be sent to the UWO email address assigned to students by Western Technology Services (WTS), i.e. your email address @uwo.ca. It is each student's responsibility to read his/her email on a frequent and regular basis.*

However, note that email at WTS (your UWO account) and other email providers such as google, hotmail.com or yahoo.com establish quotas or limits on the amount of space available to you. If you let your email accumulate there, your mailbox may fill up and you may lose important email from your instructors. Losing email is not an acceptable excuse for not knowing about the information that was sent. Similarly, emails may end up in your "Junk" mail folder for whatever reason. It is the student's responsibility that no email from the instructing team gets placed into the Junk mail folder.

- *OWL will be used to post announcements, assignments, material in the resource folder, etc. It is the student's responsibility to check OWL frequently and regularly to ensure that no course-related material is missed.*

Student contacting Staff

- For **technical issues** concerning the course (e.g., special topic, class project, domain model, summaries, etc.), students are encouraged to create Slack channels and class-wide forums to post their questions/answers for sharing their, or learning from, others' knowledge and experience. Emails to staff for this purpose will **not** be responded to.
- For **administrative issues** (e.g., absenteeism, marks, and course registration) emails to the staff are acceptable.
- Email **subject line** MUST include: "cs4471A/cs9549A:" <subject>. Without this, the email may be trapped by SPAM filter and may not be available, read or responded to.

Academic Accommodation for Medical Illness

- for work representing 10% or more of the overall grade in the course:

If you are unable to meet a course requirement due to illness or other serious circumstances, you must provide valid medical or other supporting documentation to your Dean's office as soon as possible and contact your instructor immediately. It is the student's responsibility to make alternative arrangements with their instructor once the accommodation has been approved and the instructor has been informed. In the event of a missed final exam, a "Recommendation of Special Examination" form must be obtained from the Dean's Office immediately. For further information please see:

<http://www.uwo.ca/sci/counselling/pdf/Submission-of-Medical-Documentation-for-Course-Appeal.pdf>

A student requiring academic accommodation due to illness should use the Student Medical Certificate when visiting an off-campus medical facility or request a Record's Release Form (located in the Dean's Office) for visits to Student Health Services. The form can be found here:

http://www.uwo.ca/univsec/pdf/academic_policies/appeals/medicalform.pdf

Students who are in emotional/mental distress should refer to [Mental Health@Western](#) for a complete list of options about how to obtain help.

- for work representing less than 10% of the overall grade in the course:
There are no such components in this course.

Links to the policies on Accommodation:

Link to policy on [Accommodation for Illness](#)

www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_illness.pdf
(which includes a link to the [Student Medical Certificate](#))

Link to the policy on [Accommodation for Students with Disabilities](#)

www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_disabilities.pdf

Link to the policy on [Accommodation for Religious Holidays](#)

www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_religious.pdf

[New] Western University is committed to a thriving campus as we deliver our courses in the mixed model of both virtual and face-to-face formats. We encourage you to check out the Digital Student Experience website to manage your academics and well-being:
<https://www.uwo.ca/se/digital/>.

Link to the website for Registrarial Services:

- <http://www.registrar.uwo.ca>

Link to services provided by the University Students' Council:

- <http://westernusc.ca/services/>

Accessibility Statement

You may wish to contact Services for Students with Disabilities (SSD) at 661-2111 x 82147 for any specific question regarding an accommodation.

Ethical Conduct

Scholastic offences are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offence, at the following Web site:

[http://
www.uwo.ca/univsec/pdf/academic_policies/appeals/scholastic_discipline_undergrad.pdf](http://www.uwo.ca/univsec/pdf/academic_policies/appeals/scholastic_discipline_undergrad.pdf)

Plagiarism: Students must write their essays and assignments in their own words. Whenever students take an idea, or a passage from another author, they must acknowledge their debt both by using quotation marks where appropriate and by proper referencing such as footnotes or citations. Plagiarism is a major academic offence.

You may discuss approaches to problems among yourselves; however, the actual details of the work (coding, answers to concept questions, etc.) must be an individual effort.

The standard departmental penalty for assignments that are judged to be the result of academic dishonesty is, for the student's first offence, a mark of zero for the assignment, with an additional penalty equal to the weight of the assignment also being applied. You are responsible for reading and respecting the Computer Science Department's policy on Scholastic Offences and Rules of Ethical Conduct.

The University of Western Ontario uses software for plagiarism checking. Students may be required to submit their written work and programs in electronic form for plagiarism checking.