cs4471b & cs9549b
Software Design and Architecture

Course Outline: Winter 2018

Logistics and Instruction:

Class Venue	**MC 105B**
Day and Hours	Th 10.30 AM – 1.30PM
Instructor	Nazim H. Madhavji (Madhavji <<aatt>> geee-may-l ☺)
Office Hours:	TH: 1.30 PM –2.30pm Catch me right after the class!

Sessional Dates

Term begin	Mon 8th January, 2018.
First class	Thu 11th January, 2018.
Last class	Thu 5th April, 2018.
Term end	Wed 11th April, 2018.
Reading week	Mon 19th – Fri 23th February, 2018.

Students' comments: **click here.**

Noteworthy points: **click here**

Quick Access:

Important Announcements

Important Announcements

☛ NEW/RECENT ☛ OLD	**Date**	**Description**
☛		○ **For all the course resources, please click here (restricted access): click here.**
☛	8 Jan., 2018	Please bring your laptops in class – we will need them.

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 functionalies 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.

In this course, the key work components include:

- Weekly assigned readings and creating summaries.
- Specialist topic: literature readings and in-class presentation.
- In-class Q&A and discussions.
- Team project.

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
- understanding and use of architectural tactics and patterns in the design of system architectures

- understanding and experience (through a class project) with architecture creation, analysis, and documentation.
- understanding of management and governance issues
- understanding of architecture and business
- understanding of newer architectures

Theoretical concepts learnt in the class are, in part, complemented by implementation experience in a project.

Expectations

- 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 that together form a system architecture.

  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 with various stakeholders such as user representation, requirements analysts, verification specialists, project managers, product and release managers, designers and integrators, and others.

  A system will be implemented using architectural pattern(s).

- 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.
- Lectures and 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, much “active learning” occurs outside the class time.

Prerequisites

- For undergraduate students: S3307 -- Object-Oriented Design and Analysis. For graduate students, an undergraduate degree with software engineering course(s).

- Also, please note of 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 the 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

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

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

Evaluating Software Architecture: Methods and Case Studies

Paul Clements, Rick Kazman, and Mark Klein

Addison Wesley, 2004

ISBN: 9-780201-704822

Software Architecture: Foundations, Theory, and Practice

Richard N. Taylor, Nenad Medvidovic and Eric Dashofy

Wiley, 2009

ISBN-10: 0470167742

Software Architecture, Perspectives on an emerging discipline

Mary Shaw and David Garlan

Prentice Hall, 1996

ISBN: 0-13-182957-2

Essential Software Architecture
Ian Gorton

Springer, 2006

ISBN: 978-3-540-28713-1

The art of software architecture, Design methods and Techniques

Stephen T. Albin

Wiley, 2003

ISBN: 0-471-22886-9

Large-scale software architecture, A practical guide using UML

Jeff Garland

Richard Anthony

Wiley, 2003

ISBN: 0-470-84849-9

Applied software architecture

Christine Hofmeister, Robert Nord and Dilip Soni

Addison-Wesley, 2000

ISBN: 0-201-32571-3

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 assigned based on the provided descriptions of an individual's background, skills and experience. Any adjustments in team membership to be made will be done only at the beginning of the course. 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 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 quiz, test, 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-drop period – 7th March, 2018 [NHM: Actually the last class before this is: 1st March, 2018])	15**	Weekly till 1st March, 2018
Summaries of assigned readings (Post-drop period till end of term). This subsumes the Pre-drop period mark** (max. 15%).	20	Weekly post-drop period till the end of the term.
Class presentation (formal)	20	As scheduled in class.
Questions and Answers (live) [NHM: in-class interactions/discussion]	10	Throughout the term.
System architecting project	50	5th April, 2018 (see project document for details)
Class attendance mandatory [NHM: University policy applies]	**-5% (minus 5%)** for each class missed	Attendance may be taken in the first 10 mins of the class.
Show and Tell class presentation (voluntary)	**3% BONUS**	

Email Contact

We will occasionally 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 Information Technology Services (ITS), i.e. your email address @uwo.ca. It is each student's responsibility to read this email on a frequent and regular basis, or to have it forwarded to an alternative email address if preferred. See the ITS website for directions on forwarding email.

However, note that email at ITS (your UWO account) and other email providers such as 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.

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

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

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.

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.