
Foundations of Software Engineering

Werner Dietl <wdietl@uwaterloo.ca>

Table of Contents

1. Overview	1
2. Contact	1
2.1. Teaching assistant	1
3. Course content	2
4. Course objective	2
5. Course expectations	2
6. Overview of topics	3
7. Course material	3
8. Course Schedule	4
8.1. Project	5
9. Official outline	5
10. Policies	5
11. Acknowledgments	7

1. Overview

ECE651: Foundations of Software Engineering is an introductory graduate course covering a wide range of software architecture and design topics.

2. Contact

Lectures are held **Mondays from 17:30 to 20:20 in E7 4043**¹. There are no lab or tutorial slots and students are expected to find time in open hours to complete their work.

My office hours are **by appointment** and will be held in my office, EIT 4007. I will also be available immediately after lectures. See contact details².

Course material, announcements, and submissions will be handled through Learn³.

Begin all email subjects with [ECE651].

Try not to leave your questions until the last minute.

2.1. Teaching assistant

Jenny Xiang txiang@

¹ <http://www.adm.uwaterloo.ca/cgi-bin/cgiwrap/infocour/salook.pl?level=grad&sess=1189&subject=ECE&cournum=651>

² <https://ece.uwaterloo.ca/~wdietl/contact.html>

³ <https://learn.uwaterloo.ca/d2l/home/413994>

3. Course content

Prerequisite. Graduate standing in the ECE department at Waterloo.

Programming skills in an object-oriented programming language. Experience in writing small-to-medium-sized programs (~20 kLOC) and with setting up and using large open-source projects.

Pre/Co-req: ECE 650 or 750 Tpc 26, or instructor consent.

Antireq: ECE 355, ECE 451, CS 445, CS 645, SE 463, ECE 452, CS 446, CS 646, SE 464

Grading. Grades may be curved or adjusted at the instructor's discretion.

Project: 20%, Design Assignment: 5%, Architecture Assignment: 10%, Midterm: 15%, Final exam: 50%.

You must pass the final exam, midterm, and project to pass the course.

Deliverable	Date	Format	Value
Design background	Sept 14	Learn	
Project groups	Sept 18	Learn	P/F
Proposal	Sept 21	Learn + TA	P/F
Design Assignment	Oct 5	PeerScholar	5%
Midterm	Oct 22	In Class	15%
Prototype	Oct 26	Learn + TA	P/F
Architecture Assign.	Nov 9	PeerScholar	10%
Complete Project	Dec 3	Learn + TA	20%
Final Exam	TBD	Written	50%

This is a tentative schedule that will get adapted during the term.

Details for the project and assignments will be provided separately.

No late submissions will be accepted.

4. Course objective

To introduce students to the software design process and its models. Representations of design/architecture. Software architectures and design plans. Design methods. Design state assessment. Design quality assurance. Design verification. Group design and implementation of an application.

5. Course expectations

It is expected that students attend lectures and complete the required assignments. Lectures will often include hands-on activities; participation in these exercises is essential to succeed in the class. Slides will

be provided via Learn. Any material discussed in class or in the required readings will be testable unless otherwise noted.

By the end of the course you should be able to:

- propose and analyze software architectures.
- explain the strengths and weaknesses of various architectural styles and design patterns / techniques.
- communicate and rationalize architectural and design decisions.
- ideate, justify, and implement software designs.
- evaluate, compare, and contrast different architectures and designs.

6. Overview of topics

- Software architecture, architectural styles, and architectural representations
- Software design, design patterns, design representations
- Software architecture and design conception, analysis, and communication
- Architecture and design recovery / reverse engineering
- Architecture and design visualization / understanding
- Cloud / grid computing architectures

7. Course material

While the course does not have a required textbook, much of the materials will be sourced from the following texts; additional books are supplementary.

- Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Available for purchase online⁴.
- Richard N. Taylor, Nenad Medvidovic, and Eric Dashofy. *Software Architecture. Foundations, Theory, and Practice*. Available in the library or for purchase (e.g., through Amazon.ca⁵). Slides for this book are available online⁶.

⁴ https://books.google.ca/books/about/Clean_Architecture.html?id=8ngAkAEACAAJ

⁵ [http://www.amazon.ca/gp/product/0470167742?](http://www.amazon.ca/gp/product/0470167742?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0470167742)

[ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0470167742](http://www.amazon.ca/gp/product/0470167742?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0470167742)

⁶ <http://www.softwarearchitecturebook.com/resources/>

- Ian Gorton. *Essential Software Architecture*. Available online⁷ or for purchase (e.g., through Amazon.ca⁸). Slides for this book are available online⁹.
- Fred P. Brooks Jr. *The Mythical Man Month*. Available in the library or for purchase (e.g., through Amazon.ca¹⁰).
- Fred P. Brooks Jr. *The Design of Design*. Unfortunately not in the library but still available e.g. through Amazon.ca¹¹.
- *The Architecture of Open Source Applications*. Available online¹².
- Kai Qian, Xiang Fu, Lixin Tao, Chong-wei Xu. *Software Architecture and Design Illuminated*. Available for purchase online¹³.

8. Course Schedule

All lecture material will be available through Learn¹⁴.

This is a tentative schedule that will get adapted during the term.

Week	Class
1	Introduction, organization, and project
2	Architectural views & decomposition
3	Non-functional properties, Architectural Styles
4	Architectural Styles, Architecture and Security
5	Study Day: lecture moved to Thursday, Oct. 12
5	Modeling, Design Patterns
6	Design Patterns
7	Frameworks
8	Dependency Injection, Cloud/REST Architectures
9	Design tools and verification
10	Service-oriented-Architectures, Microservices
11	Languages: Scala, Checker Framework, TypeScript

⁷ <http://bit.ly/9RF18P>

⁸ [http://www.amazon.ca/gp/product/3540287132?](http://www.amazon.ca/gp/product/3540287132?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=3540287132)

[ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=3540287132](http://www.amazon.ca/gp/product/3540287132?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=3540287132)

⁹ <https://docs.google.com/viewer?url=http://www.ug.it.usyd.edu.au/~iango/home/ESA-Slides.pdf>

¹⁰ [http://www.amazon.ca/gp/product/0201835959?](http://www.amazon.ca/gp/product/0201835959?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201835959)

[ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201835959](http://www.amazon.ca/gp/product/0201835959?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201835959)

¹¹ [http://www.amazon.ca/gp/product/0201362988?](http://www.amazon.ca/gp/product/0201362988?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201362988)

[ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201362988](http://www.amazon.ca/gp/product/0201362988?ie=UTF8&tag=s01ae8-20&linkCode=as2&camp=15121&creative=330641&creativeASIN=0201362988)

¹² <http://aosabook.org/en/index.html>

¹³ https://books.google.ca/books/about/Software_Architecture_and_Design_Illumin.html?id=aK9N6596BIIC

¹⁴ <https://learn.uwaterloo.ca/d2l/home/413994>

Week	Class
12	Project presentations, Outlook and wrap-up

8.1. Project

The project forms an integral part of this course. We will discuss details about the project in the first week of the course and adapt it to the number of students taking the course.

9. Official outline

This is the high-level outline provided by the ECE department¹⁵; this course will follow the general guideline, but will be adjusted according to your feedback, interests, and experience.

Fundamentals of software requirement analysis, software development as an engineering activity, basic process models, software specifications, modularity, cohesion, coupling, encapsulation, information hiding, principles of object oriented design, software project management, quality assurance and control.

Principles of Software Architecture: Fundamental software architecture styles, synchronous & as synchronous communication of software components. Languages for software design specification: UML (class diagrams, sequence diagrams, collaboration diagrams, state diagrams).

Overview of verification and validation techniques. Maintenance, evolution and reengineering, configuration management. Software metrics, quality assurance, fundamental cost and effort prediction models. Trends in software engineering (e.g., model-driven development, agile approaches).

Pre/Co-req: ECE 650 or 750 Tpc 26, or instructor consent.

Antireq: ECE 355, ECE 451, CS 445, CS 645, SE 463, ECE 452, CS 446, CS 646, SE 464

10. Policies

Academic Integrity

- In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. [See the academic integrity¹⁶ site for more information.]
- Turnitin.com: Text matching software (Turnitin®) will be used to screen assignments in this course. Turnitin® is used to verify that all materials and sources in assignments are documented. Students'

¹⁵ <https://uwaterloo.ca/graduate-studies-academic-calendar/node/5799>

¹⁶ <http://www.uwaterloo.ca/academicintegrity/>

submissions are stored on a U.S. server, therefore students must be given an alternative (e.g., scaffolded assignment or annotated bibliography), if they are concerned about their privacy and/or security. Students will be given due notice, in the first week of the term and/or at the time assignment details are provided, about arrangements and alternatives for the use of Turnitin® in this course. It is the responsibility of the student to notify the instructor if they, in the first week of term or at the time assignment details are provided, wish to submit the alternate assignment.

Grievance

- A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70¹⁷, Student Petitions and Grievances, Section 4.
- When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

Discipline

- A student is expected to know what constitutes academic integrity¹⁸ to avoid committing an academic offence, and to take responsibility for his/her actions.
- A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean.
- For information on categories of offences and types of penalties, students should refer to Policy 71¹⁹, Student Discipline.
- For typical penalties check Guidelines for the Assessment of Penalties²⁰.

Appeals

- A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground.
- A student who believes he/she has a ground for an appeal should refer to Policy 72²¹, Student Appeals.

Note for Students with Disabilities

- AccessAbility Services²², located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without

¹⁷ <http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm>

¹⁸ <http://www.uwaterloo.ca/academicintegrity/>

¹⁹ <http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm>

²⁰ <http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm>

²¹ <http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm>

²² <https://uwaterloo.ca/disability-services/>

compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the AccessAbility Services at the beginning of each academic term.

11. Acknowledgments

Thanks to Derek Rayside, Reid Holmes, and Krzysztof Czarnecki for sharing their experience and materials from previous iterations of this and a similar course, CS 446, Winter 2014²³.

²³ <https://cs.uwaterloo.ca/~rtholmes/teaching/2014winter/cs446/index.html>