
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	2
7. Course material	3
7.1. Schedule	3
7.2. Project	4
8. Official outline	4
9. Policies	5
10. Acknowledgments	6

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 **Tuesdays from 17:30 to 20:20 in DWE 2527**¹. There are no lab or tutorial slots.

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

TBD.

¹ <http://www.adm.uwaterloo.ca/cgi-bin/cgiwrap/infocour/salook.pl?level=grad&sess=1169&subject=ECE&cournum=651>

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

³ <https://learn.uwaterloo.ca/d2l/home/270124>

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.

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

Participation: 5%, Project: 25%, Midterm: 20%, Final exam: 50%.

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

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 first two texts; additional books are supplementary.

- 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.ca⁴). Slides for this book are available online⁵.
- Ian Gorton. *Essential Software Architecture*. Available online⁶ or for purchase (e.g., through Amazon.ca⁷). Slides for this book are available online⁸.
- Fred P. Brooks Jr. *The Mythical Man Month*. Available in the library or for purchase (e.g., through Amazon.ca⁹).
- Fred P. Brooks Jr. *The Design of Design*. Unfortunately not in the library but still available e.g. through Amazon.ca¹⁰.

7.1. Schedule

All lecture material will be available through Learn¹¹.

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

Week	Date	Class
1	Sept. 13	Introduction, organization, and project

⁴ [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)

⁵ <http://www.softwarearchitecturebook.com/resources/>

⁶ <http://bit.ly/9RF18P>

⁷ [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)

⁸ <https://docs.google.com/viewer?url=http://www.ug.it.usyd.edu.au/~iango/home/ESA-Slides.pdf>

⁹ [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)

¹⁰ [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)

¹¹ <https://learn.uwaterloo.ca/d2l/home/270124>

Week	Date	Class
2	Sept. 20	Architectural views & decomposition
3	Sept. 27	Non-functional properties, Architectural Styles
4	Oct. 4	Architectural Styles, Architecture and Security
5	Oct. 11	Design 1
6	Oct. 18	Design 2
7	Oct. 25	Midterm
8	Nov. 1	Dependency Injection, Cloud/REST Architectures
9	Nov. 8	Design tools and verification
10	Nov. 15	Frameworks
11	Nov. 22	Project presentations
12	Nov. 29	Outlook and wrap-up

7.2. 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.

8. Official outline

This is the high-level outline provided by the ECE department¹²; 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 & asynchronous 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).

Antireq: ECE 355, 451, 452

¹² <http://www.ucalendar.uwaterloo.ca/SA/GRAD/1516/GRDcourse-ECE.html#ECE651>

9. 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 integrity¹³ site for more information.]
- Turnitin.com: Text matching software (Turnitin®) will be used to screen assignments in this course. This is being done to verify that use of all materials and sources in assignments is documented. Students will be given an option if they do not want to have their assignment screened by Turnitin®. In the first week of the term, details will be provided about arrangements and alternatives for the use of Turnitin® in this course.

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 70¹⁴, 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 integrity¹⁵ 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 71¹⁶, Student Discipline.
- For typical penalties check Guidelines for the Assessment of Penalties¹⁷.

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.

¹³ <http://www.uwaterloo.ca/academicintegrity/>

¹⁴ <http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm>

¹⁵ <http://www.uwaterloo.ca/academicintegrity/>

¹⁶ <http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm>

¹⁷ <http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm>

- A student who believes he/she has a ground for an appeal should refer to Policy 72¹⁸, Student Appeals.

Note for Students with Disabilities

- AccessAbility Services¹⁹, located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without 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.

10. Acknowledgments

Thanks to Derek Rayside and Reid Holmes for sharing their experience and materials from previous iterations of similar courses, in particular CS 446, Winter 2014²⁰.

¹⁸ <http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm>

¹⁹ <https://uwaterloo.ca/disability-services/>

²⁰ <https://cs.uwaterloo.ca/~rtholmes/teaching/2014winter/cs446/index.html>