

---

# Software Reliability Engineering

Werner Dietl <wdietl@uwaterloo.ca>

## Table of Contents

1. Overview .....	1
2. Contact .....	2
2.1. Teaching assistant .....	2
3. Course content .....	2
3.1. Prerequisite .....	2
3.2. Material .....	2
3.3. Schedule .....	2
3.4. Grading .....	3
3.5. Course Policies .....	4
4. Official outline .....	4
5. Policies .....	4
6. Acknowledgments .....	6

## 1. Overview

**ECE654: Software Reliability Engineering** is an **advanced** graduate course. After introducing basic concepts and models for software reliability and security, we will delve into state-of-the-art research and available tools. It is expected that you will read and prepare research papers that will be assigned for the different lectures and that you experiment with the different research prototypes.

The lectures will consist of presentations by the lecturer and the students. Individual projects will complement the lectures and will give first-hand experience with current research.

Topic areas:

- Unit testing, automatic test case generation
- Bug finding tools: Findbugs, PMD, ...
- Coding guidelines for reliable and secure software
- Type systems; The Checker Framework
- Design-by-Specification: Spec#, OpenJML, ...
- Advanced verification approaches: Dafny, rise4fun, ...
- Model checking: Spin, Java Path Finder, NuSVM, ...
- Theorem proving: Coq, Isabelle, ...

## 2. Contact

Lectures are held asynchronously online<sup>1</sup>. There are no lab or tutorial slots.

My office hours are **by appointment** and will be held in MS Teams. See contact details<sup>2</sup>.

Course material, announcements, and submissions will be handled through Learn<sup>3</sup>.

Begin all email subjects with [ECE654].

Try not to leave your questions until the last minute.

### 2.1. Teaching assistant

TBD.

## 3. Course content

### 3.1. Prerequisite

Graduate standing in the ECE department at Waterloo.

Advanced programming skills in an object-oriented programming language. Experience in writing medium-sized programs (>20k LOC) and with setting up and using large open-source projects.

Prereq: ECE 650 or 750 Tpc 26 or instructor consent.

Coreq: ECE 650 or 750 Tpc 26 or instructor consent.

Antireq: ECE 750 Topic 2.

### 3.2. Material

There is no required textbook that needs to be bought. Lecture slides, lecture notes, and reading material will be made available on Learn.

### 3.3. Schedule

All lecture material will be available through Learn<sup>4</sup>.

This is a tentative schedule that will get adapted during the term. Check Learn for updates.

---

<sup>1</sup> <https://classes.uwaterloo.ca/cgi-bin/cgiwrap/infocour/salook.pl?level=grad&sess=1215&subject=ECE&cournum=654>

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

<sup>3</sup> <https://learn.uwaterloo.ca/d2l/home/684204>

<sup>4</sup> <https://learn.uwaterloo.ca/d2l/home/684204>

Week	Date	Class
1	May 6	Introduction and organization
2	May 13	Basic reliability and security concepts and models
3	May 27	Testing, bug finding, and coding guides
4	June 3	Static analysis, lattices, type systems
5	June 10	Quiz 1; Soundness, Progress, Preservation
6	June 17	Object-oriented type systems
7	June 24	Design-by-Specification
8	July 2	Model checking, Theorem proving
9	July 8	Quiz 2; Symbolic Execution
10	July 15	Security fundamentals
11	July 22	Language-based security
12	July 29	Outlook and wrap-up

The dates for the quizzes might change. Dates will be finalized at least a week before the respective quiz.

## 3.4. Grading

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

Assignments: 30%

Quizzes: 20%

Final Exam: 50%

All assignments and quizzes will be returned in class, during office hours, or through LEARN.

You must pass the final exam to pass the course. You must pass the weighted average of quizzes and final to pass the course. The final grade will be computed as follows:

```
def grade(assignments, quizzes, final):
    normal = 0.3 * assignments + 0.2 * quizzes + 0.5 * final
    weighted = (0.2 * quizzes + 0.5 * final) / 0.7
    if final < 50:
        return min(normal, final)
    elif weighted < 50:
        return min(normal, weighted)
    else:
        return normal
```

If you missed the quizzes (and had permissions from the instructor according to the Course Policy), the final grade will be computed as follows:

```
def grade(assignments, quizzes, final):
    normal = 0.3 * assignments + 0.7 * final
    if final < 50:
```

```
return min(normal, final)
else:
return normal
```

## 3.5. Course Policies

By registering for this class, students agree to the following class policies:

**Independent work.** All work turned in will be that of the individual student unless stated otherwise. Violations would result in zero credit to all students concerned. Policy 71 will be followed for any discovered cases of plagiarism.

**Lateness.** You have 2 days of lateness to use on assignment submissions throughout the term. Each day you hand in an assignment late consumes one of the days of lateness. If you consume all of your late days, assignments that are still late will get 0 marks. You can only hand in an assignment up to the time all assignments are returned. Missed assignments get 0 marks. For example, you may hand in A1 two days late and A2 on time, or you can hand in A1 one day late and A2 one day late.

**Missed Quizzes.** If you miss a quiz, you will receive 0 marks for the quiz. If you have a legitimate reason (at the discretion of the instructor) that you cannot take a quiz, and obtain permission from the instructor a week in advance, the percentage for the quiz may be shifted to the final. No alternative quiz time will be provided.

## 4. Official outline

This is the high-level outline provided by the ECE department<sup>5</sup>; this course will follow the general guideline, but will be adjusted according to your feedback, interests, and experience.

The course consists of two related parts. The first part deals with the engineering of reliable software. It introduces basic software reliability concepts, describes relevant models and discusses processes for engineering of reliable software, including schemes and patterns for the design of reliable and fault tolerant software. The second part addresses development of secure software. It presents key software security concept, techniques and models, overviews major software security vulnerabilities and their exploitation, and considers processes for development of secure software.

## 5. 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 integrity<sup>6</sup> site for more information.]

---

<sup>5</sup> <https://uwaterloo.ca/graduate-studies-academic-calendar/node/7158>

<sup>6</sup> <http://www.uwaterloo.ca/academicintegrity/>

- 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' 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 70<sup>7</sup>, 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 integrity<sup>8</sup> 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 71<sup>9</sup>, Student Discipline.
- For typical penalties check Guidelines for the Assessment of Penalties<sup>10</sup>.

## 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 72<sup>11</sup>, Student Appeals.

## Note for Students with Disabilities

- AccessAbility Services<sup>12</sup>, located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without

---

<sup>7</sup> <http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm>

<sup>8</sup> <http://www.uwaterloo.ca/academicintegrity/>

<sup>9</sup> <http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm>

<sup>10</sup> <http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm>

<sup>11</sup> <http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm>

<sup>12</sup> <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.

## **6. Acknowledgments**

Thanks to Mohammad Ahmad Munawar and Rudolph E. Seviora for sharing their course material for previous iterations of the course.