EECS 3311 3.00: Software Design

Section Z, Winter 2022

Instructor

- Song Wang (`http://www.eecs.yorku.ca/~wangsong`)
- Contact: wangsong@yorku.ca
- Lecture Times: 14:30 PM – 16:00 PM, Tuesdays, Thursdays;
- Zoom: https://yorku.zoom.us/my/wangsonging
- Office Hours: 16:00 PM – 19:00 PM, Tuesdays; or by Appointments.

Online Submission/Assessment

- All announced deadlines are in the Eastern Time Zone (Toronto time). Students on a different time zone can figure out the corresponding local time.
- Stringent deadlines are imposed on all quizzes, lab assignments, projects. A final is scheduled online (via eClass) with stringent timing requirements (start time, duration, and end time to be announced by the registrar office).
- Students are responsible for taking proactive steps and/or seeking assistance well in advance to ensure that their technical setup (e.g., reasonably stable internet connection, a working computer which does not freeze sporadically) allows them to complete and submit each online assessment item (quiz, lab, project, exam) in time.

Prerequisites

- **General Prerequisites**: A cumulative grade point average (GPA) of 4.50 or better over all previously completed Major EECS courses. The GPA computation excludes all EECS courses that have a second digit 5, or are Co-Op/PEP courses.
- LE/EECS 2011 3.00
- LE/EECS 2031 3.00
- SC/MATH 1090 3.00

Course Description

A study of design methods and their use in the correct construction, implementation, and maintenance of software systems. Topics include software life cycles, software design, software implementation, software testing, documentation needs and standards, support tools. Students design and implement components of a software system.

This course focuses on design techniques for both small and large software systems. Techniques for the design of components (e.g., modules, classes, procedures, and executables) as well as complex architectures will be considered. Principles for software design and rules for helping to ensure software quality will be discussed. The techniques will be applied in a set of small assignments, and a large-scale project, where students will design, implement, and maintain a non-trivial software system.

Three lecture hours and 1.5 lab hours, weekly.

Course Learning Outcomes

Upon completion of the course, students are expected to be able to:

CLO1 Implement specifications with designs that are correct, efficient, and maintainable.

CLO2 Develop systematic approaches to organizing, writing, testing, and debugging software.

CLO3 Develop insight into the process of moving from an ambiguous problem statement to a well-designed solution.

CLO4 Design software using appropriate abstractions, modularity, information hiding, and design patterns.

CLO5 Develop facility in the use of an IDE for editing, organizing, writing, debugging, documenting designs, and the ability to deploy the software in an executable form.

CLO6 Describe software specifications via Design by Contract, including the use of preconditions, postconditions, class invariants, as well as loop variants and invariants.

CLO7 Write precise and concise software documentation that also describes the design decisions and why they were made.

Reference Textbooks

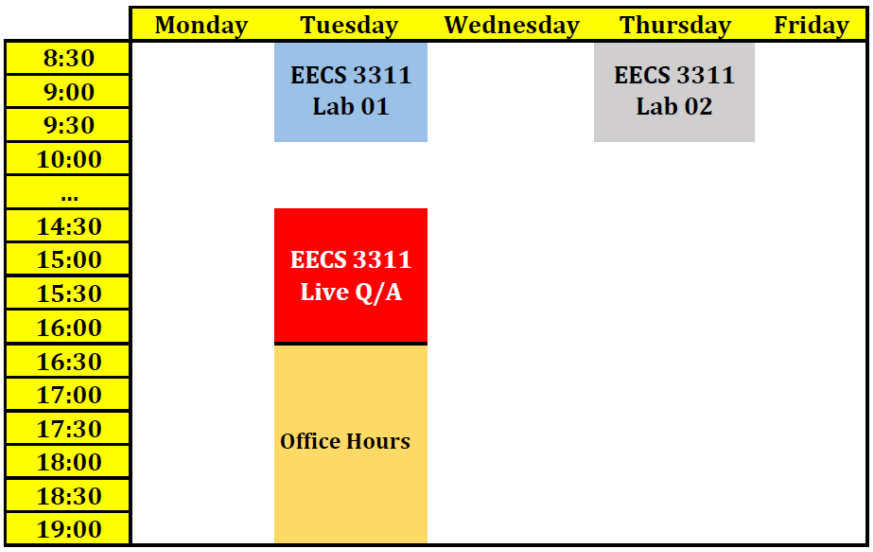
1 **Title**: Design Patterns: Elements of Reusable Object-Oriented Software
Author: Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
Publisher: Addison Wesley, 1994.
Edition: First Edition
ISBN-10: 0201633612
ISBN-13: 978-0201633610

2 **Title**: UML Distilled: A Brief Guide to the Standard Object Modeling Language
Author: Martin Fowler
Publisher:Addison-Wesley, 2003.
Edition: Third Edition
ISBN-13: 9780321193681

3 **Title**: Domain Driven Design: Tackling Complexity in The Heart of Software
Author: Eric Evans
Publisher: Addison-Wesley, 2014.
Edition: First Edition
ISBN-13: 9780321125217

4 **Title**: Touch of Class: Learning to Program Well with Objects and Contracts
Author: Bertrand Meyer
Publisher: Springer, 2013.
Edition: Second Edition
ISBN-13: 9783540921448

Scheduled Q&A and Lab Sessions



- Videos of lectures will be pre-recorded and shared in ECHO360 on Sundays.
- Lectures on Tuesdays (14:30 PM – 16:00 PM) will be held as live Q&A sessions to answer your questions related to the lecture videos.
- Labs will be live weekly for tutorial/QA related to labs/project.
- You are welcome to attend any of the scheduled lab or Q&A sessions.

Zoom for Live Q/A and Labs: https://yorku.zoom.us/my/wangsonging

Academic Integrity

- All labs and the project are to be completed **individually**: no group work is allowed.
- All lab and project submissions will be check automatically via plagiarism checkers: suspicious submissions will be reported to Lassonde for a formal investigation.
- To protect yourself from ending up a submission that is suspiciously similar to someone else's, you want to avoid:
  - Discussing code-level details about labs/project with anyone.
  - Discussing concrete steps about your solution or someone's solution.
  - Sharing any part(s) of your solutions.
  - Giving or receiving instructions about what exactly you should type for a fragment of code.
  - **It is acceptable to ask about how to solve a question in general (i.e., how to write a loop in general), but unacceptable to ask about how to write code specifically for solving a problem.**

Reporting Cases

Enforcing the policy of academic honesty not only maintains the standard of the course, but also ensures fairness among all students in the class. If you have sufficient reasons to believe that cases of violation are present, let the instructor know and confidentiality will be maintained.

Quizzes

- Quizzes will be released after the lectures on **Thursdays** and they cover topics introduced in previous lectures.
- Quizzes are open-booked.
- Each quiz will be opened for its submission for 48 hours.
- Each quiz will consist of between 5 and 10 questions with a variety of forms (e.g., multiple choice, matching answers, true/false).
- There is only one single attempt allowed for the quiz.
- All quizzes are on **https://eclass.yorku.ca/eclass/**

Grading Scheme (tentative)

Component	Out	Due	Percentage
Lab0	Jan 11st	-	0%
Lab1	Jan 18th	Jan 31st	10%
Lab2	Feb 1st	Feb 14th	10%
Lab3	Feb 15th	March 7th	10%
Project	March 8th	April 8th	15%
Weekly Quizzes	Thursdays	-	10%
Midterm	March. 8th	-	15%
Final Exam	TBD	-	30%

Letter Grades and their Interpretations

Letter Grade	Grade Point	Interpretation
A+	9	Exceptional
A	8	Excellent
B+	7	Very Good
B	6	Good
C+	5	Competent
C	4	Fairly Competent
D+	3	Passing
D	2	Marginally Passing
E	1	Marginally Failing
F	0	Failing

– For each grading unit you are assigned a raw mark score.

– The raw mark score is not a grade as it is merely used to **rank** you in the class.

– You will be provided with a mapping from your raw mark score to a letter grade.

e.g., A raw mark score of 78 (out of 100) might be a C, not a B+, after the mapping is applied.

e.g., A raw mark score of 40 (out of 100) might be a D, not an E, depending on the mapping.

Tasks in Weekly Lab Sessions (topics are tentative)

Week	Date	Task
Week 1	Jan. 11st	Lab0: Eclipse, Maven, Github, Junit
Week 2 Week 3	Jan. 18th Jan. 25th	Lab1: Polymorphism & Dynamic Binding
Week 4 Week 5	Feb. 1st Feb. 8th	Lab2: Design Patterns
Week 6	Feb. 15th	Lab3: Design by Contract
Reading Week (Feb 19th – 25th)		
Week 7	March 1st	Lab3: Design by Contract
Week 8 Week 9	March. 8th March. 15th	Project (Software Design)
Week 10 Week 11 Week 12	March. 22nd March. 29th April. 5th	Project (GUI-based Software Implementation)

Tentative Course Calendar

Week	Day	Date	Topics
1	T	11/01	Introduction; Software Process Models
	R	13/01	Basic OOP Design Principles
2	T	18/01	Basic OOP Design Principles
	R	20/01	UML
3	T	25/01	Write Good Test Cases
	R	27/01	Singleton and Factory Patterns
4	T	01/02	Builder and Prototype Patterns
	R	03/02	Adapter Pattern
5	T	08/02	Bridge and Composite Pattern
	R	10/02	Visitor and Observer Patterns
6	T	15/02	Iterator & State Patterns
	R	17/02	Software Architectural Models

Winter Reading Week: Feb. 19th – 25th			
Week	**Day**	**Date**	**Topics**
7	T	01/03	DbC (Java without tool support)
	R	03/03	JML Supported DbC
8	T	08/03	Midterm Test
	R	10/03	DbC Practice
9	T	15/03	Test Automation
	R	17/03	Random Test Case Generation
10	T	22/03	Search-based Test Generation
	R	24/03	Testing Management
11	T	29/03	Testing Management
	R	31/03	Software Document
12	T	05/03	Software Verification
	R	07/03	Wrap-Up
Winter Study Day: April 11st			
Exam Period: April 12nd – 29th			