EECS 3311 3.00: Software Design

Winter 2024

Instructor

- Song Wang (`http://www.eecs.yorku.ca/~wangsong`)
- Contact: wangsong@yorku.ca
- Lecture Times: 16:00 PM – 17:30 PM, Mondays and Wednesdays;
- Office Hours: 14:00 PM – 16:00 PM, Mondays; or by Appointments.

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**: Clean Architecture: A Craftsman's Guide to Software Structure and Design Paperback
Author: Robert Martin
Publisher: Pearson, 2017.
Edition: 1st Edition
ISBN-13: 9780134494166

Labs

- Lab Sec. 01

  10:00 am – 11:30 am on Mondays

  Venue: Lassonde Building (LAS) 1006 [D5/19 on the Keele Campus Map]

- Lab Sec. 02

  17:30 pm – 19:00 pm on Wednesdays

  Venue: Lassonde Building (LAS) 1006 [D5/19 on the Keele Campus Map]

Attendance

- The range of topics covered in this course is extensive, and due to the limited lecture and lab time, these topics are covered in an intensive manner. Therefore, attendance at both lectures and labs are *necessary* in order for you to keep up and perform well.

- Students are responsible for attending all classes and lab sessions, arriving on time, and coming fully prepared to discuss the assigned readings and exercises.

Lab Tests

- In chosen lab sessions, you will be required to complete programming tasks (using an IDE) or writing tasks. These tests are designed to test your understanding of the taught concepts, as well as your mastery of using the programming tool to develop working solutions to given problems.

- These tests are based on lecture materials and lab exercises. For your preparation, instructions of the lab test will be distributed in advance.

- For your submission to be assessed, you **must** submit *compilable* source code. We will use auto marking tool to examine your written code, so you receive very low marks by submitting code that does not compile.

- Lab test to be done individually in the lab.

Academic Integrity

- On default, all labs 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.**

Quizzes

- We have around four quizzes. A quiz will be released every other week on **Fridays** and they cover topics introduced in the lectures in between two quizzes.
- Quizzes are open-booked.
- Each quiz will be opened for its submission for 24 hours.
- Each quiz will consist of around 10 questions with a variety of forms (e.g., multiple choice, matching answers, true/false, or written questions).
- There is only one single attempt allowed for the quiz.
- All quizzes are on **https://eclass.yorku.ca/eclass/.**

Late Submission Policy for Projects

We accept late submissions of the projects, the policy is as follows:

- Submitted on time before the deadlines (or extended deadlines): Whatever mark it gets from the Rubric.
- Within the next 24hours: 10% penalty.
- Within the next 48hour: 50% penalty.

Grading Scheme (tentative)

Component	Out	Due	Percentage
Lab Test 1	the week of Feb. 5th	-	5%
Lab Test 2	the week of March. 11th	-	7%
Lab Test 3	the week of April. 1st	-	8%
Project Deliverable 1	Jan. 16th	Feb. 18th	10%
Project Deliverable 2	-	March. 18th	10%
Project Deliverable 3	-	April. 1st	15%
Quiz 1	Jan. 26th	in 24 hours	2.5%
Quiz 2	Feb. 9th	in 24 hours	2.5%
Quiz 3	March. 8th	in 24 hours	2.5%
Quiz 4	March. 22nd	in 24 hours	2.5%
Final Exam	TBD	-	35%

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

Tasks in Weekly Lab Sessions (topics are tentative)

Week	Date	Task
Week 1	Jan. 8th	Basic tools: Eclipse, Github, Junit
Week 2	Jan. 15th	Class Diagrams
Week 3	Jan. 22nd	Class Diagrams
Week 4	Jan. 29th	**Lab Test 1**
Week 5	Feb. 5th	Design Patterns
Week 6	Feb. 12th	Design Patterns
Reading Week (Feb. 18th – 24th)		
Week 8	Feb. 26th	Software Architecture Design
Week 9	March. 4th	**Lab Test 2**
Week 10	March. 11th	Code Smell
Week 11	March. 18th	DBC
Week 12	March. 25th	DBC
Week 13	April. 1st	**Lab Test 3**

Tentative Course Calendar

Week	Topics
1	Introduction; Basic OOP Design Principles Junit
2	UML UML
3	SOLID Principles Design Pattern (Creational Design Pattern)
4	Pattern (Structural Pattern) Design Pattern (Structural Pattern)
5	Design Pattern (Behavioral Patterns) Architectural Patterns
6	Architectural Patterns Process Models

Winter Reading Week: Feb. 17th – 23rd	
Week	**Topics**
7	TDD Test Case
8	Input Space For Tests Input Space For Tests
9	Test Automation Testing Management
10	Code Smells/Refactoring Code Smells/Refactoring
11	Static Analysis Static Analysis
12	Static Tools JML Supported DbC
13	Wrap-Up & Review
Winter Study Day: April. 9th **Exam Period: April. 10th – 27th**	