

San José State University
Engineering Extended Studies

CMPE 202
Software Systems Engineering

Section 47
Spring 2024

Course and Contact Information

| | |
|------------------|--|
| Instructor: | Ron Mak |
| Office location: | Clark Hall CL 325 (mostly working from home) |
| Email: | ron.mak@sjsu.edu |
| Website: | http://www.cs.sjsu.edu/~mak/ |
| Office hours: | TuTh 4:30 – 5:30 PM in person, Clark Hall CL 325 |
| Class days/time: | Tu: 6:00 – 8:45 PM |
| Classroom: | TBA |
| Prerequisites: | Classified graduate standing or instructor consent. Artificial Intelligence or Computer Engineering or Software Engineering majors only. |

Course Catalog Description

“Integrated approach to software design and development including requirements elicitation and analysis, system design and construction through studying multiple facets of software development processes, design methodologies, modeling approaches, and implementation techniques.”

Course Format

This class will meet in person in the classroom. Exams will be given in the classroom.

Faculty Web Page and Canvas

Course materials, syllabus, assignments, grading criteria, exams, and other information will be posted at my [faculty website](http://www.cs.sjsu.edu/~mak) at <http://www.cs.sjsu.edu/~mak> and on the [Canvas Learning Management System course login website](http://sjsu.instructure.com) at <http://sjsu.instructure.com>. You are responsible for regularly checking these websites to learn of any updates. You can find Canvas video tutorials and documentations at <http://ges.sjsu.edu/canvas-students>.

Course Goals

The primary goal of this course is to become a much better programmer.

The instructor will share decades of experience as a successful software developer in industry and in scientific research institutions. The programming examples will be in C++, but the material will apply well to other object-oriented languages.

Course Learning Outcomes (CLO)

Upon successful completion of this course, students will be able to:

- CLO 1: **Software design and development process:** Use an approach to software development consisting of multiple design-code-test iterations. Model design with UML diagrams.
- CLO 2: **Requirements gathering and analysis:** Gather the requirements for a software application, distinguish between functional and nonfunctional requirements, and express the requirements in the form of use cases. Analyze the requirements to derive the initial set of classes.
- CLO 3: **Design principles:** Practice good design principles, including single responsibility, open-closed, code to the interface, Law of Demeter, and Liskov Substitution Principle.
- CLO 4: **Recursion and backtracking:** Design elegant solutions to certain types of programming problems using powerful recursion and backtracking techniques.
- CLO 5: **Design patterns:** Learn major “Gang of Four” design patterns and recognize when it is appropriate to apply them.

Recommended Texts

| | |
|--|---|
| Title: Author: Publisher: ISBN: <u>Book website:</u> | Object-Oriented Software Design in C++ Ronald Mak Manning, 2024 978-1633439504 https://www.manning.com/books/object-oriented-software-design-in-c-plus-plus?ar=true&lpse=A The class will be based on this, the instructor’s new book (still in prepublication — see the book website). |
| Title: Author: Publisher: ISBN: | Design Patterns: Elements of Reusable Object-Oriented Software Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides Addison-Wesley Professional, 1994 978-0201633610 The original “Gang of Four” design patterns book. |
| Title: Author: Publisher: ISBN: | UML Distilled: A Brief Guide to the Standard Object Modeling Language, 2nd edition Martin Fowler Addison-Wesley, 2004 0-321-19368-7 This book teaches the Unified Modeling Language (UML). |

Software to Install

To do GUI-based programming in C++, download and install the **Qt Framework** of graphical user interface library classes: <https://www.qt.io/product/framework>.

We will use **Qt Widgets**, and so also download and install **Qt Designer**:
<https://build-system.fman.io/qt-designer-download>

Course Requirements and Assignments

You should have good C++ programming skills and be familiar with use an interactive development environment (IDE) such as Eclipse CDT, CLion, Visual Studio, Microsoft Visual C++, or Apple Xcode. Install and use your preferred IDE.

You will work during the semester in small teams. Programming assignments will provide practice with good software design.

Each team will also have a semester design project to develop an application that it can demonstrate to the class. Each team will write a short report (10-15 pp.) that describes the design principles and patterns that it used, including a high-level architecture description with UML diagrams.

Each team will submit its assignments and project into Canvas, which will display the scoring rubrics for grading. At the end of the semester, each team will give a presentation and demo of its design project, and students will help to score each presentation.

Each assignment and project will be worth up to 100 points. Late assignments will lose 20 points and an additional 20 points for each 24 hours after the due date.

The university's syllabus policies:

- [University Syllabus Policy S16-9](http://www.sjsu.edu/senate/docs/S16-9.pdf) at <http://www.sjsu.edu/senate/docs/S16-9.pdf>.
- Office of Graduate and Undergraduate Program's [Syllabus Information web page](http://www.sjsu.edu/gup/syllabusinfo/) at <http://www.sjsu.edu/gup/syllabusinfo/>

“Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.”

Technology Requirements

Students are required to have an electronic device (laptop, desktop, or tablet) with a camera and microphone. SJSU has a free [equipment loan program](https://www.sjsu.edu/learnanywhere/equipment/index.php) available for students:
<https://www.sjsu.edu/learnanywhere/equipment/index.php>

Students are responsible for ensuring that they have access to reliable Wi-Fi during tests. If students are unable to have reliable Wi-Fi, they must inform the instructor, as soon as possible or at the latest one week before the test date to determine an alternative. See [Learn Anywhere](#) website for current Wi-Fi options on campus.

Grading Information

Individual total scores will be computed with these weights:

| | |
|-----|-----------------|
| 35% | Assignments* |
| 30% | Design project* |
| 15% | Midterm exam** |
| 20% | Final exam** |

* *team scores*

** *individual scores*

Programming assignments will be scored on how well the programs were written based on rubrics to be posted to Canvas for each assignment.

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Program's [Syllabus Information web page](http://www.sjsu.edu/gup/syllabusinfo/) at <http://www.sjsu.edu/gup/syllabusinfo/>.

CMPE 202

Software Systems Engineering

Section 47
Spring 2024
Instructor: Ron Mak

Course Schedule (subject to change with fair notice)

| Week | Date | Topics |
|------|--------|---|
| 1 | Jan 30 | Overview of the course The path to well-designed software Review of object-oriented software design Requirements to build the right application The Functional Specification document |
| 2 | Feb 6 | Good class design to build the application right Introduction to Unified Modeling Language (UML) diagrams Use cases Unit testing and regression testing The Design Specification document Software verification and validation (V&V) |
| 3 | Feb 13 | Hide class implementations The Principle of Least Knowledge The Open-Closed Principle Code to the Interface Principle Rules from the Law of Demeter Don't surprise users |
| 4 | Feb 20 | Design subclasses right Function overriding vs. overloading "Is-a" vs. "has-a" relationships The Liskov Substitution Principle |
| 5 | Feb 27 | Recursion and backtracking Binary search trees Quicksort The Eight Queens puzzle The Sudoku puzzle |
| 6 | Mar 5 | <i>Midterm exam (75 minutes)</i> GUI-based programming |
| 7 | Mar 12 | The Template Method Design Pattern The Strategy Design Pattern |
| 8 | Mar 19 | The Factory Method Design Pattern The Abstract Factory Design Pattern |
| 9 | Mar 26 | The Adapter Design Pattern The Façade Design Pattern |
| | | <i>Spring break, April 1 – April 5</i> |

| Week | Date | Topics |
|------|-------------------|--|
| 10 | Apr 9 | The Iterator Design Pattern The Visitor Design Pattern |
| 11 | Apr 16 | The Observer Design Pattern |
| 12 | Apr 23 | The State Design Pattern |
| 13 | Apr 30 | The Singleton Design Pattern The Composite Design Pattern The Decorator Design Pattern |
| 14 | May 7 | Team project presentations |
| | Tuesday May 21 | <i>Final exam</i> 5:15 – 7:30 PM Classroom TBA |