

Object-Oriented Design

CS 151

Fall 2026 Section 01 In Person 3 Unit(s) 08/19/2026 to 12/07/2026 Modified 08/19/2026

Contact Information

- **Instructor:** Dr. Suneuy Kim
- **Office:** MacQuarrie Hall 217 (MH217)
- **Telephone:** 408-924-5122
- **Email:** suneuy.kim@sjsu.edu (<mailto:suneuy.kim@sjsu.edu>)
Preferred mode of contact is email.
- **Office Hours:** Tuesday and Thursday, 10:30 AM–11:30 AM in MH217 or on Zoom [Meeting ID: 838 0090 4104](#)

When you email me with a question, use **[Q]** in the subject line to receive a reply within a reasonable response time. For example:

Subject: [Q] lecture note

Course Information

- **Course:** CS151, Object-Oriented Design
- **Section:** 1
- **Semester:** Fall 2026
- **Instruction:** In Person
- **Class Days/Time/Classroom:** TuTh 9:00 AM - 10:15 AM, Duncan Hall 450

Class Format: Lectures are conducted **in person**; Zoom is used only for the **poll at the end** of most classes. The Zoom meeting (Meeting ID: [873 9750 6563](#)) is accessed through Canvas. **Students must register for the Zoom meeting for their poll responses to be recorded.** None of these sessions will be recorded.

Course Description and Requisites

Design of classes and interfaces. Object-oriented design methodologies and notations. Design patterns. Generics and reflection. Exception handling. Concurrent programming. Graphical user interface programming. Software engineering concepts and tools. Required team-based programming assignment.

Prerequisite(s): MATH 42, CS 46B, and [(CS 48 or CS 49J) if CS 46B was not in Java], each with a grade of C- or better; Allowed Declared Majors: Computer Science, Applied and Computational Math, Software Engineering, or Data Science; or instructor consent.

Grading: Letter Graded

* Classroom Protocols

Academic Integrity

- Any cheating on an exam will result in a grade of F in the class.
- If duplicate programs are found, both the provider and the copier will receive 0 points on the assignment. A second offense results in a grade of F in the class.
- Any incident of academic dishonesty will be reported to the University for disciplinary action.

Attendance

University Policy F15-12 states that:

"Students should attend all meetings of their classes, not only because they are responsible for material discussed therein, but because active participation is frequently essential to insure maximum benefit for all members of the class. Attendance per se shall not be used as a criterion for grading."

Consent for Recording of Class and Public Sharing of Instructor Material

University Policy S12-7 requires students to obtain the instructor's permission to record the course.

Common courtesy and professional behavior dictate that you notify someone when you are recording them. You must obtain the instructor's permission to make audio or video recordings in this class. Such permission allows the recordings to be used for private study purposes only. The recordings are the intellectual property of the instructor; you have not been given any rights to reproduce or distribute the material.

Course material cannot be shared publicly without the instructor's approval. You may not publicly share or upload instructor-generated material for this course, such as exam questions, lecture notes, or homework solutions, without instructor consent.

☰ Program Information

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

🎯 Course Goals

- OO Design:
 - Introduce core UML concepts
 - Introduce a simplified OO analysis and design methodology
 - Present the concept of design pattern
 - Present the concept of a software framework

- Java Language:
 - Make students proficient in the use and creation of interfaces and inheritance hierarchies
 - Make students proficient in the Java type system
 - Introduce threads and thread safety
- Software Engineering:
 - Introduce a GUI toolkit, including basic widgets and the event handling mechanism
 - Introduce basic software engineering concepts and tools

Course Learning Outcomes (CLOs)

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

- OO Design
 - Interpret and produce UML class diagrams and UML sequence diagrams
 - Develop simple use cases, perform noun-verb analysis, interpret and produce CRC cards
 - Appropriately select and apply key design patterns in the construction of a software application
 - Be able to follow a systematic OO design methodology
- Java language
 - Create a class hierarchy involving existing and new interfaces and classes, including inner classes.
 - Design, implement, test, and debug programs in an object-oriented language, involving the creation of at least 10 classes and interfaces
 - Use generic types, reflection, and lambda expressions
 - Implement concurrent programs and use thread-safe data structures
- Software Engineering
 - Use a GUI toolkit to create a graphical user interface involving frames, buttons, text components, panels, menus, and simple geometric shapes
 - Be able to document use cases for a simple team project
 - Be able to plan and track a simple team project
 - Be able to use a version control system and an automated build system

Course Materials

Required Texts/Readings

- Textbook: Object-Oriented Design & Patterns, 3rd edition, by Cay Horstmann

Course Requirements and Assignments

Programming Assignments and Project

- Four individual programming assignments involving design and implementation.
- All submitted programs should be original. For some assignments, you are allowed to borrow a code from the textbook. If this is the case, it will be clearly specified in the assignment description and you have to acknowledge it in the corresponding source code.
- A group project with 3 members per group in the last month of the semester involves OO design and GUI programming. A project description and guidelines will be posted later.
- All assignments will be submitted through Canvas.

Submission/Late Policy

- All submissions must be completed through Canvas. Email submissions will not be accepted for grading.
- Any assignments/project turned in past the deadline will get a penalty: For each late day, a 20% of the maximum obtainable score of the work will be taken out of what you earned. A late day is one 24-hour period beyond the due date.
- For example, suppose the maximum score of an assignment is 100 and you earned 80 points. If the submission is late by two days, the final score of the assignment would be $80 - 2 \times 20 = 40$.
- Any submission turned in more than 48 hours past the deadline will result in a grade of zero for that assignment.
- You can submit your work multiple times. If then, the latest one will be considered as the final submission. If the final submission is late, the late policy will be applied.

Teamwork Policy

- Once a team is formed, it will last throughout the semester. If you dissolve your team, a significant amount of penalty will be determined by the instructor and given to both parties.
- For the project, students are expected to report their own results as well as their collaborators.
- The task responsibility and contribution of every team member must be precisely documented in a report.
- Team members will be graded individually based on the report and peer evaluation.

Programming Language

- Get the latest long-term support release of Java SE Platform.
- Download: <https://www.oracle.com/java/technologies/downloads/>
(<https://www.oracle.com/java/technologies/downloads/>).

UML Tools

- Students may use **Lucidchart** to create UML diagrams for course assignments. Lucidchart is available through the **SJSU One.SJSU portal**. Log in to <https://one.sjsu.edu/> (<https://one.sjsu.edu/>) with your SJSU credentials and access Lucidchart from the available applications. Lucidchart may be used to create UML class diagrams, sequence diagrams, use-case diagrams, and other diagrams required for assignments.
- **Violet**: <http://horstmann.com/violet> (<http://horstmann.com/violet>).
- You may use a UML tool of your choice.

IDE

- IntelliJ IDEA: <http://jetbrains.com/idea> (<http://jetbrains.com/idea>).
- Eclipse: <http://eclipse.org/> (<http://eclipse.org/>).

Testing

- [JUnit 5 \(https://junit.org/junit5/?utm_source=chatgpt.com\)](https://junit.org/junit5/?utm_source=chatgpt.com) will be used for basic unit testing in Java programming assignments.

Technology Requirements

Students are required to have an electronic device (laptop, desktop or tablet) with a camera and built-in microphone.

SJSU has a free **equipment loan** program available for students:

<https://www.sjsu.edu/learnanywhere/equipment/index.php>

[. \(https://www.sjsu.edu/learnanywhere/equipment/index.php\)](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 at the latest one week before the test date.

See the **Learn Anywhere** website for current Wi-Fi options on campus:

<https://www.sjsu.edu/learnanywhere/equipment/index.php>

[. \(https://www.sjsu.edu/learnanywhere/equipment/index.php\)](https://www.sjsu.edu/learnanywhere/equipment/index.php)

Evaluation

Exams

- There will be two midterm exams and one comprehensive final exam.
- All exams will be given online via Canvas.
- The dates of midterm exams will be announced with fair notice.
- The final exam date is scheduled by the university and cannot be changed.
- See the semester schedule at the end of the syllabus for the final exam schedule.
- All Canvas exams for this class must be taken **in the classroom** during the **scheduled class time for midterms** or the **scheduled exam time for the final**.
- Exams taken remotely will **not** be accepted for credit.

Makeup Exam Policy

Absolutely no make-up exams will be offered under any circumstances.

For those who couldn't take the exam or worked hard but had a bad day on the exam day ending up with a low score, I offer the following opportunity to possibly replace your worst midterm score with the final score.

If your final exam (percentage) grade is higher than your worst midterm (percentage) grade, then I will replace the worst midterm grade with your final exam grade.

For example, if you have a 60% on your worst midterm and you receive an 80% on the final exam, I will replace the 60% by 80% in the computation of your course grade.

Online Exams

Proctoring Software and Exams

Exams will be proctored in this course through LockDown Browser.

✓ Grading Information

You will receive the final grade based on the weighted average score on your performance. The grading weights are as follows.

Evaluation Component	Weight
Exam I	20%
Exam II	20%
Final Exam	35%
Programming Assignments	15%
Project	8%
Participation	2% (poll in class)
Total	100%

Regrading

If you believe an error was made in the grading of your assignment or exam, you may request a regrade by speaking with the instructor during office hours or by sending me an email. Regrade requests must be made **within one week** of the date the assignment or exam grade is posted.

Letter Grade Determination

First, letter grades are normally assigned according to the following percentage scale:

Grade	Percentage
A+	97.0–100.0
A	93.0–96.9

Grade	Percentage
A-	90.0–92.9
B+	87.0–89.9
B	83.0–86.9
B-	80.0–82.9
C+	77.0–79.9
C	73.0–76.9
C-	70.0–72.9
D+	67.0–69.9
D	63.0–66.9
D-	60.0–62.9
F	0.0–59.9

However, if the overall class performance is too low for these standard cutoffs to be appropriate, the instructor may adjust the grading scale.

In that case:

1. The instructor will establish a C- cutoff that is below the overall class average but above 60%. This cutoff may be adjusted as needed.
2. Students whose overall scores are at or above the C- cutoff will be assigned letter grades from C- to A+ based on their performance relative to the class.
3. Students whose scores are below the C- cutoff will receive D+, D, D-, F, or WU, depending on their performance relative to the class.
4. The same grading method applies to all students enrolled in the course, including graduate students.

In short: the standard percentage scale will be used when appropriate. If overall class performance is significantly lower than expected, the instructor may use a curved grading system, with the C- cutoff serving as the dividing line between the higher letter grades and the lower grades.



University Policies

Per [University Policy S16-9 \(PDF\)](http://www.sjsu.edu/senate/docs/S16-9.pdf), relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance,

counseling, and other resources) are listed on the [Syllabus Information](https://www.sjsu.edu/curriculum/courses/syllabus-info.php) (<https://www.sjsu.edu/curriculum/courses/syllabus-info.php>) web page. Make sure to visit this page to review and be aware of these university policies and resources.

Course Schedule

Subject to change with fair notice at least one class period in advance. Students will be notified in class and/or via Canvas should any changes occur.

Week	Topics
1	Introduction to CS151
1	Object-Oriented Design Process
2	Object-Oriented Design Process
2	Object-Oriented Design Process
3	Object-Oriented Design Process
3	Object-Oriented Design Process
4	Interface Types and Polymorphism
4	Interface Types and Polymorphism
5	Interface Types and Polymorphism
5	Interface Types and Polymorphism
6	Interface Types and Polymorphism
6	Interface Types and Polymorphism
7	Lambda Expression
7	Patterns and GUI Programming
8	Midterm I
8	Patterns and GUI Programming
9	Patterns and GUI Programming
9	Patterns and GUI Programming
10	Patterns and GUI Programming
10	Inheritance and Abstract Classes
11	Inheritance and Abstract Classes
11	Inheritance and Abstract Classes
12	Midterm II
12	The SOLID Principles
13	The Java Object Model
13	The Java Object Model
14	The Java Object Model
14	Concurrent Programming
15	Concurrent Programming (Last Day of Instruction)
Final Exam	Thu, December 10, 8:30-10:30 AM