

San José State University
College of Science/Department of Computer Science
CS151, Object-Oriented Design, Section 4 & 5, Spring, 2020

Course and Contact Information

Instructor(s):	Katarzyna Tarnowska
Office Location:	MacQuarrie Hall (MQH) 215
Telephone:	(408) 9245076
Email:	katarzyna.tarnowska@sjsu.edu
Office Hours:	Mondays & Wednesdays 3 PM – 4 PM
Class Days/Time:	Monday & Wednesday, 4:30 PM – 5:45 PM (S4)/1:30PM – 2:45PM (S5)
Classroom:	MQH222 (S4)/DH450 (S5)
Prerequisites:	Math 42, CS46B, and CS 49J (or equivalent knowledge of Java) with a grade of C- or better in each or instructor consent.

Course Description

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.

Course Format

Course Web Page

Course materials such as syllabus, handouts, notes, assignment instructions, etc. will be published on Canvas Learning Management System course login website at <https://sjsu.instructure.com/courses/1360438>. You are responsible for regularly checking with the Canvas system to learn of any updates. For help with using Canvas see Canvas Student Resources page (http://www.sjsu.edu/ecampus/teaching-tools/canvas/student_resources)

Course Goals

Introduce students to the basic principles of OO Design, plus elements of UML and design patterns. Cover the Java language features not yet seen in CS1 and CS2. Teach basic GUI programming.

- 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 (CLO)

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

1. 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 the following design patterns in the construction of a software application: Composite, Decorator, Iterator, Strategy, Template method, and Observer
 - Be able to follow a systematic OO design methodology
2. 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
 - Throw, propagate and catch exceptions
 - Implement concurrent programs and use thread-safe data structures
3. 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

Required Texts/Readings

Textbook

C. Horstmann. Object-Oriented Design & Patterns, 3rd edition.

Other Readings

Java Swing, 2nd ed., O'Reilly.

Other technology requirements / equipment / material

- Programming Language: [Java Platform SE 8 or later](#)
 - [Download](http://www.oracle.com/technetwork/java/javase/downloads/index.html) at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- IDE:
 - [Eclipse](http://eclipse.org/) at <http://eclipse.org/>
- Version control (for group projects):
 - [Git](https://git-scm.com/) at <https://git-scm.com/>
 - [Git for Eclipse](https://www.eclipse.org/egit/) at <https://www.eclipse.org/egit/>
- UML design (choice of one):
 - Lucidchart
 - Violet
 - SAP Power Designer

Course Requirements and Assignments

- [University Policy S16-9](#): “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.”
- There will be four programming assignments, at least 2 weeks in duration, involving design and implementation.
- Team-based project. Design and implement an application (topics will be given later in the semester). Apply object-oriented design process (CRC and UML), GUI design & programming. Event-driven implementation. Documentation and collaboration tools required.
- Late policy: there will be late penalty applied for a late submission as provided in the description of the assignment.
- Extra points may be assigned for participation in class.

Final Examination or Evaluation

Pre-requisite, midterm and final exams will be closed book with multiple choice multiple answers questions, and short-answer open-ended questions.

Grading Information

Determination of Grades

- Total points for the course will be weighted by:
 - Entry exam 10% 100 points
 - Midterm exam 20% 200 points
 - Final exam 20% 200 points
 - Programming assignments 20% 200 points
 - Project 30% 300 points
 - Analysis & Design 10% 100 points
 - Implementation 20% 200 points
- Letter grades will be assigned according to the following policy:

1000– 960----A+
959 – 910 ---- A
909 -- 860 ---- A-
859-- 810---- B+
809 -- 760 ---- B
759 -- 710 ---- B-
709 -- 660 ---- C+
659 -- 610 ---- C
609 -- 560 ---- C-
559 -- 510 ---- D+
509 -- 460 ---- D
459 -- 410 ---- D-
409 -- 00 ---- F

Classroom Protocol

- Attendance: students should attend all meetings of their classes ([University Attendance and Participation Policy F15-12](#)).
- Arrival: students are expected to arrive on time. Late students might not be admitted to the class.
- Behavior: eating, cellphone use, personal loud discussions are not allowed in the classroom. Laptops permitted as per instructions.
- Communication: use your official SJSU e-mail account to communicate with the instructor. Messages sent through e-mail domains other than sjsu.edu will be discarded.
- Any violation of the academic integrity will result in 0 points for the assignment in which the violation occurred and reporting to the Office of Student Conduct and Ethical Development.

University Policies

- Policy on Academic Integrity
“Your commitment, as a student, to learning is evidenced by your enrollment at San Jose State University. The [University Academic Integrity Policy F15-7](#) requires you to be honest in all your academic course work. Faculty members are required to report all infractions to the office of Student Conduct and Ethical Development. Visit the [Student Conduct and Ethical Development](#) website for more information.”

Per [University Policy S16-9](#) (<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 [Syllabus Information web page](#) (<http://www.sjsu.edu/gup/syllabusinfo>), which is hosted by the Office of Undergraduate Education. Make sure to visit this page to review and be aware of these university policies and resources.

CS151 / Object-Oriented Design, Spring 2020, Course Schedule

Course Schedule (tentative)

The schedule is subject to change with fair notice.

Week	Date	Topics, Readings, Assignments, Deadlines
1	1/27	Introduction to the course, A Crash Course in Java, Chapter 1
1	1/29	Entry exam.
2	2/3	The Object-Oriented Design Process, Chapter 2
2	2/5	The Object-Oriented Design Process, Chapter 2.
3	2/10	The Object-Oriented Design Process, Chapter 2
3	2/12	The Object-Oriented Design Process, Chapter 2
4	2/17	Guidelines for Class Design, Chapter 3
4	2/19	Guidelines for Class Design, Chapter 3
5	2/24	Guidelines for Class Design, Chapter 3
5	2/26	Design Patterns, Chapter 5.
6	3/2	Design Patterns, Chapter 5
6	3/4	Design Patterns, Chapter 5
7	3/9	Design Patterns, Chapter 5
7	3/11	Interfaces and Polymorphism, Chapter 4
8	3/16	Interfaces and Polymorphism, Chapter 4
8	3/18	Interfaces and Polymorphism, Chapter 4
9	3/23	Interfaces and Polymorphism, Chapter 4
9	3/25	Midterm
10	3/30	Spring Recess – Campus Closed
10	4/1	Spring Recess – Campus Closed
11	4/6	Inheritance and Abstract Classes, Chapter 6
11	4/8	Inheritance and Abstract Classes, Chapter 6
12	4/13	Inheritance and Abstract Classes, Chapter 6
12	4/15	Inheritance and Abstract Classes, Chapter 6
13	4/20	The Java Object Model, Chapter 7
13	4/22	The Java Object Model, Chapter 7

Week	Date	Topics, Readings, Assignments, Deadlines
14	4/27	Frameworks, Chapter 8
14	4/29	Concurrency, Chapter 9
15	5/4	Concurrency, Chapter 9
15	5/6	Concurrency, Chapter 9
16	5/11	Final exam
17	5/19 5/18	14:45-17:00 (S4) – Project presentations 12:15-14:30 (S5) – Project presentations