San José State University
College of Science/Department of Computer Science
CS151, Object-Oriented Design, Section 8, Fall, 2021

Course and Contact Information

Instructor(s):	Mariia Surmenok
Office Location:	Online via zoom: https://sjsu.zoom.us/j/7233378568?pwd=Q2ZCNFBXelVjRHZJR0Vxb0JpU3FMZz09
Email:	Mariia.surmenok@sjsu.edu
Office Hours:	Tuesdays 3:30 PM – 4:30 PM, Thursdays 1:15 PM – 2:15 PM
Class Days/Time:	Tuesdays & Thursdays 2:15 PM – 3:30 PM
Classroom:	Online via zoom: https://sjsu.zoom.us/j/86901860682?pwd=OXNRY0w1Vk5IODR1ZXhSSTR1TVdldz09
Prerequisites:	MATH 42, and (CS 46B in Java or (CS 49J and CS 46B)) with a grade of C- or better in each

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

Technology Intensive, Online Course

Faculty Web Page and MYSJSU Messaging

We will use Canvas and Slack chat. Also, we will use iClicker for polling during the lectures.

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
- 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 technology requirements / equipment / material

- Programming Language: Java Platform SE 8 or later http://www.oracle.com/technetwork/java/javase/downloads/index.html
- IDE: You are free to use any IDE, I will use Intellij IDEA from JetBrains. You can download community edition for free or use your university email to get ultimate version https://www.jetbrains.com/idea/download/
- Version control (for assignments and group project): git at https://git-scm.com/ and GitHub account to store code remotely
- 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."*

Exams (30%)
One midterm exam (15%) and final exam (15%). Exams cannot be made up, except for reasons of illness, as certified by a doctor, or documentable extreme emergency.
We will use Proctorio extension with Google Chrome to take midterm and final exams + you need a web camera.

Quizzes (20%)
Quizzes will be given throughout the course covering the required material discussed. Quizzes will be given every Tuesday (or biweekly if there are not enough material) on Canvas and should be completed by 11:59 PM of Wednesday. Two lowest scores will be dropped. No make-up quizzes.

Programming Assignments (20%)
Approximately 3-5 homework assignments. Code should be uploaded to GitHub and document (for example, Readme.pdf) with GitHub link and any written answers should be uploaded on Canvas.
Late Homework
Schedule your time well to protect yourself against unexpected problems.

- One late homework without penalty for up to 7 days. If using this option, should email to me, so I can notify grader to not place any deduction. Cannot be combined with late homework (below). Would not be accepted 7 days after the original due date.
- All other late works is accepted with a penalty of 10% per day. Late homework is not accepted one week past it's due date All homework is due at 11:59PM (Pacific Time) on the due date specified.

Project (30%)
One team project with 3 people per team. One team leader per project. Project will involve OO design and GUI programming. **Project should include task planning and tasks assignment for each team member**. The project details will be posted in separate document. The deadline for each project milestone can be found in Course Schedule section below.
At the end of the project, each team member should submit the report explaining his/her own contribution and peer evaluation describing the contribution of your team members.

Extra credit
The iClicker participation points will be used for extra credits for Exams

Grading Information

Your grade for the course is based on the exams, the homework, and quizzes. Grades are calculated by weighting the scores as defined below. For each section (exams, quizzes, homeworks) the grade cannot exceed 100% even if you get extra points for this section.

A+ will be given to the top 3 students with score 99 or above

At least	Letter Grade
93	A
90	A-
87	B+
83	B
80	B-
77	C+
73	C
70	C-
67	D+
63	D
60	D-
Below 60	F

Classroom Protocol

Exams:

- The exams are based on lectures, homework/lab assignments, and reading materials covered before the exam's date.
- NO collaboration or usage of someone else's help including the web sites like with solutions (chegg.com, etc)

Individual work:

- All homework and exams must be your own individual work. It is OK to have general discussions about the assignments or read other material for inspiration. You may never copy anything from anyone without attribution. This means if you find code on Stackoverflow or another web site, you need to give the URL where you found the code in a comment at the top of your class so that I can look at it if necessary.
- You may copy from the textbook, the labs, or anything we do in class without attribution. For assignments and exams, you may not copy anything from any other student at all, and you may not collaborative produce results in pairs or teams. Your work must be entirely your own.
- It is never okay to share your code with other students. If the other person submits your work, both students will receive a 0.
- **First incident of cheating will result in a 0. Second incident will result in a F for the class.**

University Policies

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 2021, Course Schedule

Course Schedule

The schedule is subject to change with fair notice.

August 31 – Last day to drop without W
September 8 – Last day to add

Week/Lesson /Module	**Date**	**Topics, Readings, Assignments, Deadlines** ***(If appropriate, add extra column(s) to meet your needs.)***
1	08/19	Introduction to the course. A Crash Course in Java, Chapter 1
2	08/24	A Crash Course in Java, Chapter 1 **Quiz 1 (Diagnostic quiz)**
2	08/26	Command line basics, Git and GitHub. **Homework 1 out**
3	08/31	The Object-Oriented Design Process, Chapter 2 **Quiz 2**
3	09/02	The Object-Oriented Design Process, Chapter 2 **Homework 1 due** **Project: Deadline to form groups (due Sunday 09/05 via Google Sheets)**
4	09/07	The Object-Oriented Design Process, Chapter 2 **Quiz 3**
4	09/09	Guidelines for Class Design, Chapter 3
5	09/14	Guidelines for Class Design, Chapter 3 **Quiz 4**
5	09/16	Generics, Chapter 7 **Project Milestone 1: Idea and Visual Design (due Sunday 9/19)**
6	09/21	Interfaces and Polymorphism, Chapter 4 **Quiz 5**
6	09/23	Interfaces and Polymorphism, Chapter 4
7	09/28	Interfaces and Polymorphism, Chapter 4 **Quiz 6**
7	09/30	Interfaces and Polymorphism, Chapter 4 **Project Milestone 2: System Design, Diagrams (due Sunday 10/03)**
8	10/05	Design Patterns, Chapter 5 **Quiz 7**
8	10/07	Design Patterns, Chapter 5
9	10/12	Design Patterns, Chapter 5 **Quiz 8**
9	10/14	Design Patterns, Chapter 5 **Project Milestone 3: Create Tickets and show intermediate progress on "Jira" board and GitHub (due Sunday 10/17)**

Week/Lesson /Module	**Date**	**Topics, Readings, Assignments, Deadlines** ***(If appropriate, add extra column(s) to meet your needs.)***
10	10/19	More Design Patterns, Chapter 10
10	10/21	**Midterm**
11	10/26	Inheritance and Abstract Classes, Chapter 6
11	10/28	Inheritance and Abstract Classes, Chapter 6 **Project Milestone 4: Model and incorporated Design Pattern (due Sunday 10/31)**
12	11/02	Inheritance and Abstract Classes, Chapter 6 **Quiz 9**
12	11/04	Inheritance and Abstract Classes, Chapter 6
13	11/09	The Java Object Model, Chapter 7 **Quiz 10**
13	11/11	**Veteran's Day – No Class** **Project Milestone 5: Create Tickets and show intermediate progress on "Jira" board and GitHub (due Sunday 11/14)**
14	11/16	The Java Object Model, Chapter 7 **Quiz 11**
14	11/18	Frameworks, Chapter 8
15	11/23	Concurrency, Chapter 9 **Quiz 12**
15	11/25	**Thanksgiving Holiday – No Class**
16	11/30	Concurrency, Chapter 9 **Project Milestone 6: Final Submission (due Tuesday 11/30)**
16	12/02	**Project: Presentations (volunteers, for extra points)** **Project: Submit Peer Review (individually)**
Final Exam		TBD