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):	Mariia Surmenok
Office Location:	Online via zoom (link on canvas and slack)
Email:	Mariia.surmenok@sjsu.edu
Office Hours:	Tuesdays & Thursdays 10:30 AM – 11:30 AM
Class Days/Time:	Tusdays & Thursdays 7:30 AM – 9 AM (Section 8) 9:00 AM – 10:15 AM (Section 2)
Classroom:	Online via zoom (link on canvas and slack)
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

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/
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.

Quizzes (10%)
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 the same day. Two lowest scores will be dropped. No make-up quizzes.

Programming Assignments (30%)
Approximately 4-6 homework assignments. Code should be uploaded to GitHub and readme 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 5 days. If using this option, should email to me, so I can notify grader to not place any deduction.
>
> 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 on the due date specified.

Project (30%)
Analysis and Design 10%
Implementation 20%
One team project with 3 people per team. One team lead per project. Project will involve OO design and GUI programming. The project details will be posted later.
At the end of the project, each team member should submit the report explaining its own contribution and peer evaluation describing the contribution of your team members.

Extra credit
The iClicker participation points may be used to give your final grade a slight boost.

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.

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 with solutions.

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 on that assignment or exam. 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, Fall 2020, Course Schedule

Course Schedule

The schedule is subject to change with fair notice.

Week/Lesson /Module	Date	Topics, Readings, Assignments, Deadlines *(If appropriate, add extra column(s) to meet your needs.)*
1	08/20	Introduction to the course. Git and GitHub
2	08/25	Git and GitHub. A Crash Course in Java, Chapter 1 **Quiz 1**
2	08/27	The Object-Oriented Design Process, Chapter 2
3	09/01	The Object-Oriented Design Process, Chapter 2
3	09/03	The Object-Oriented Design Process, Chapter 2
4	09/08	The Object-Oriented Design Process, Chapter 2
4	09/10	Guidelines for Class Design, Chapter 3
5	09/15	Guidelines for Class Design, Chapter 3
5	09/17	Guidelines for Class Design, Chapter 3
6	09/22	Design Patterns, Chapter 5
6	09/24	Design Patterns, Chapter 5
7	09/29	Design Patterns, Chapter 5
7	10/01	Design Patterns, Chapter 5
8	10/06	Interfaces and Polymorphism, Chapter 4
8	10/08	Interfaces and Polymorphism, Chapter 4
9	10/13	Interfaces and Polymorphism, Chapter 4
9	10/15	Interfaces and Polymorphism, Chapter 4
10	10/20	Midterm Review **No quiz**
10	10/22	**Midterm**
11	10/27	Inheritance and Abstract Classes, Chapter 6
11	10/29	Inheritance and Abstract Classes, Chapter 6
12	11/03	Inheritance and Abstract Classes, Chapter 6
12	11/05	Inheritance and Abstract Classes, Chapter 6
13	11/10	The Java Object Model, Chapter 7
13	11/12	The Java Object Model, Chapter 7

14	11/17	Frameworks, Chapter 8
14	11/19	Concurrency, Chapter 9
15	11/24	Concurrency, Chapter 9
15	11/26	**Thanksgiving Holidays- Campus Closed**
16	12/01	Project Presentations
16	12/03	Project Presentations, Final Review
Final Exam		Online