

San José State University
College of Science/Department of Computer Science
CS151, Object-Oriented Design, Section 2, Fall, 2019

Course and Contact Information

Instructor:	Dr. Katarzyna Tarnowska
Office Location:	MacQuarrie Hall (MH) 215
Telephone:	(408) 9245076
Email:	Katarzyna.tarnowska@sjsu.edu
Office Hours:	Monday & Wednesday, 3PM – 4PM
Class Days/Time:	Monday & Wednesday, 10:30PM – 11:45AM
Classroom:	MacQuarrie Hall 225
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 Format

Technology Intensive, Hybrid, and Online Courses

Students are required to bring their laptops to the class or borrow laptops from the library (check Student Technology Resources at <http://www.sjsu.edu/gup/syllabusinfo/#StudentTechResources>).

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/1323649>. You are responsible for regularly checking with the Canvas system to learn of any updates.

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. Prerequisite: MATH 42, CS 46B, and CS 49J (or equivalent knowledge of Java) (with a grade of "C-" or better in each); Computer Science, Applied and Computational Math or Software Engineering majors only; or instructor consent.

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 (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. Online copy made available on Canvas for the personal use of students enrolled in this section of the course - you must not share your copy with others.

Other technology requirements / equipment / material

- Hardware: laptops in the classroom are required.
- Software
 - Programming Language: [Java Platform SE 8](#)
 - [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):
 - SVN at <https://svnkit.com>
 - SVN for Eclipse at <https://www.eclipse.org/subversive/>
 - Git at <https://git-scm.com/>
 - Git for Eclipse at <https://www.eclipse.org/egit/>
 - UML design (choice of one):
 - MS Visio
 - 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.”*
- Team project. Design and implement an application (topics will be given later in the semester). Apply object-oriented design process (CRC and UML), design patterns, GUI programming and concurrency (detailed requirements will be presented at the lecture). Documentation and collaboration tools required.
- Pop-up quizzes/coding assignment. Students are required to prepare in the assigned textbook chapters and code. Students have to bring laptop computers to each meeting. If not instructed, laptops need to remain closed. Use of computer in the class for purposes other than education is strictly prohibited. Pop-up quizzes check the understanding of the material before or after the lecture. Coding assignments will be done in class and uploaded on Canvas. Sample such assignments will ask for short modifications of the textbook code.

Final Examination or Evaluation

Midterm and final exams will be closed book with multiple choice questions, short open-ended questions and design problems to solve.

Grading Information

Determination of Grades

- Total points for the course will be weighted by:

○ Pre-requisite exam	10%	10 points
○ Midterm exam	20%	20 points
○ Final exam	20%	20 points
○ Project: Analysis	5%	5 points
○ Project: Design	5%	5 points
○ Project: Implementation	30%	30 points
○ Pop-up quizzes/ coding assignments	10%	10 points
- Letter grades will be assigned according to the following policy:
100 -99 ---- A+
93 – 98 ---- A
89 -- 92 ---- A-
87 -- 88 ---- B+
83 -- 86 ---- B
80 -- 82 ---- B-
77 -- 79 ---- C+
73 -- 76 ---- C
70 -- 72 ---- C-
67 -- 69 ---- D+
63 -- 66 ---- D
60 -- 62 ---- D-
0 -- 59 ---- F
- No late work accepted or make-ups.

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.
- Behavior: eating, personal loud discussions are not allowed in the classroom.
- 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.”

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 Programs' [Syllabus Information web page](#) at <http://www.sjsu.edu/gup/syllabusinfo/>.

CS151 / Object-Oriented Design, Fall 2019, Course Schedule

Course Schedule

The schedule is subject to change with fair notice.

Week	Date	Topics, Readings, Assignments, Deadlines
1	8/21	Introduction to the course
2	8/26	A Crash Course in Java, Chapter 1
2	8/28	Pre-requisite exam. Intro to IDE (Eclipse).
3	9/2	Labor Day - Campus Closed
3	9/4	The Object-Oriented Design Process, Chapter 2. Project team formation.
4	9/9	The Object-Oriented Design Process, Chapter 2
4	9/11	The Object-Oriented Design Process, Chapter 2
5	9/16	The Object-Oriented Design Process, Chapter 2. Project Analysis Phase due
5	9/18	Guidelines for Class Design, Chapter 3
6	9/23	Guidelines for Class Design, Chapter 3
6	9/25	Guidelines for Class Design, Chapter 3
7	9/30	Patterns and GUI Programming, Chapter 5. Project Design Phase due.
7	10/2	Patterns and GUI Programming, Chapter 5
8	10/7	Patterns and GUI Programming, Chapter 5
8	10/9	Patterns and GUI Programming, Chapter 5
9	10/14	Midterm
9	10/16	Interfaces and Polymorphism, Chapter 4
10	10/21	Interfaces and Polymorphism, Chapter 4
10	10/23	Interfaces and Polymorphism, Chapter 4
11	10/28	Interfaces and Polymorphism, Chapter 4
11	10/30	Inheritance and Abstract Classes, Chapter 6
12	11/4	Inheritance and Abstract Classes, Chapter 6
12	11/6	Inheritance and Abstract Classes, Chapter 6
13	11/11	Veteran's Day (Observed) - Campus Closed
13	11/13	Concurrent Programming, Chapter 9
14	11/18	Concurrent Programming, Chapter 9

Week	Date	Topics, Readings, Assignments, Deadlines
14	11/20	Concurrent Programming, Chapter 9
15	11/25	The Java Object Model, Chapter 7
15	11/27	Non-Instructional Day
16	12/2	The Java Object Model, Chapter 7
16	12/4	Frameworks, Chapter 8
17	12/9	Frameworks, Chapter 8. Project Implementation Phase due.
17	12/12	Final exam MH 225, 9:45AM-12:00