San José State University
College of Science / Department of Computer Science
CS 151 Object-Oriented Design, Section 6, Spring 2017

Course and Contact Information

Instructor:	Angus Yeung, Ph.D.
Office Location:	DH 282
Email:	fai.yeung@sjsu.edu
Office Hours:	Sat 1:15 PM – 3:15 PM, and by Appointment
Class Days/Time:	Sat 10:00 AM – 12:45 PM
Classroom:	MH 422
Prerequisites:	Math 42, CS 46B, and CS 49J (or equivalent knowledge of Java) (with a grade of "C-" or better in each) or instructor consent.

Faculty Web Page and MYSJSU

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on the Canvas learning management system course website. You are responsible for regularly checking with the messaging system through MySJSU (or other communication system as indicated by the instructor) to learn of any updates.

Course Description

Design of classes and interfaces. Value and reference semantics. Object-oriented design methodologies and notations. Design patterns. Reflection and serialization. Exception handling. Graphical user interface programming. Frameworks and components. Multithreading. Required team-based programming assignment.

Learning Outcomes and 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.

Course Goals:

- Object-Oriented Design (OOD)
    - Core UML concepts
    - Simplified Object-Oriented (OO) analysis and design methodology
    - Concept of design pattern
    - Concept of a software framework
- Java Languages
    - Proficiency in the use and creation of interfaces and inheritance hierarchies
    - Proficiency in the Java type system
    - Threads and thread safety

- GUI Programming
    - GUI toolkit, including basic widgets and the event handling mechanism

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 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
- 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
    - Implement correctly the `equals`, `hashCode`, `clone`, `toString` methods
    - Use serialization, reflection, and generics
    - Throw, propagate and catch exceptions
    - Implement threads and thread-safe data structures
- GUI Programming
    - Use a GUI toolkit to create a graphical user interface involving frames, buttons, text components, panels, menus, and simple geometric shapes

BS in Computer Science Program Outcomes Supported:

These are the BSCS Program Outcomes supported by this course:

- An ability to apply knowledge of computing and mathematics to solve problems
- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
- An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
- An ability to function effectively on teams to accomplish a common goal
- An ability to use current techniques, skills, and tools necessary for computing practice
- An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
- An ability to apply design and development principles in the construction of software systems of varying complexity

Course Topics

Chapter	# Of Classes	Comments
2. Object-Oriented Design Process	2	Identifying classes and responsibilities. CRC cards, UML class, sequence, and state diagrams. Design Documentation. Case study: Analyzing a program with multiple interacting classes.
3. Guidelines for Class Design	2	Encapsulation, cohesion and coupling. Programming by contract. Unit testing. Case studies: 1) Analyzing the quality of a set of standard library classes, 2) Modifying the private implementation of a class.
4. Interface Types and Polymorphism	1	Interface types. Polymorphism. Anonymous classes. Designing an interface type. Case studies: 1) The Icon Interface type, 2) The Comparable and Comparator types.
5. Patterns and GUI Programming	2	The pattern concept. The Iterator, Observer, Strategy, Composite, and Decorator patterns. Case studies: 1) Patterns in the standard library, 2) Putting patterns to work in a large program.
6. Inheritance and Abstract Classes	2	Inheritance, abstract classes. The Template Method pattern. Protected interfaces. Case studies: 1) Inheritance in the standard library, 2) Inheritance and GUI programming, 3) Inheritance and exception handling.
7. The Java Object Model	2	The Java type system, type inquiry. The toString, equals, and hashCode methods. Shallow and deep copying. Serialization. Reflection. Generic classes and methods, wildcard types
8. Frameworks	1	The framework concept. Case studies: 1) The applet framework, 2) The collections framework.
9. Multithreading	2	Running threads. Race conditions, synchronization. Avoiding deadlocks. Case study: Algorithm animation.
10. More Design Patterns	1	The adapter pattern, Actions and the command pattern, the factory method pattern, the proxy pattern, the Singleton pattern and the visitor pattern.
	15	

Required Texts/Readings

Textbook

C. Horstmann, *Object-Oriented Design and Patterns*, 2nd ed., John Wiley & Sons, Inc., ISBN 0-471-74487-5

Other Readings

K. Arnold, J. Gosling and D. Holmes, *The Java Programming Language*, 4th Ed., Addison-Wesley.

M. Fowler, *UML Distilled*, 3rd Ed., Addison-Wesley.

E. Gamma et al., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.

X. Jia, *Object-Oriented Software Development Using Java: Principles, Patterns, and Frameworks,* 2nd Ed., Addison-Wesley.

Course Requirements and Assignments

SJSU classes are designed such that in order to be successful, it is expected that students will spend a minimum of forty-five hours for each unit of credit (normally three hours per unit per week), including preparing for class, participating in course activities, completing assignments, and so on. More details about student workload can be found in University Policy S12-3 at http://www.sjsu.edu/senate/docs/S12-3.pdf.

There will be three assignments involving design and implementation. An individual class project involves Object-Oriented Design and GUI programming.

NOTE that University policy F69-24 at http://www.sjsu.edu/senate/docs/F69-24.pdf 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."

Grading Policy

The percentage weight assigned to class assignments, class project and final exam are listed as below:

Assignment #1	10%
Assignment #2	10%
Assignment #3	10%
Assignment #4	10%
Final Project	25%
Final Exam	35%
	100%

Late or missed work will not be accepted.

Note that "All students have the right, within a reasonable time, to know their academic scores, to review their grade-dependent work, and to be provided with explanations for the determination of their course grades." See University Policy F13-1 at http://www.sjsu.edu/senate/docs/F13-1.pdf for more details.

University Policies

Dropping and Adding

Students are responsible for understanding the policies and procedures about add/drop, grade forgiveness, etc. Refer to the current semester's Catalog Policies section at http://info.sjsu.edu/static/catalog/policies.html. Add/drop deadlines can be found on the current academic year calendars document on the Academic Calendars webpage at http://www.sjsu.edu/provost/services/academic_calendars/. The Late Drop Policy is available at http://www.sjsu.edu/aars/policies/latedrops/policy/**.** Students should be aware of the current deadlines and penalties for dropping classes.

Information about the latest changes and news is available at the Advising Hub at http://www.sjsu.edu/advising/.

Consent for Recording of Class and Public Sharing of Instructor Material

University Policy S12-7, http://www.sjsu.edu/senate/docs/S12-7.pdf, requires students to obtain instructor's permission to record the course:

"Common courtesy and professional behavior dictate that you notify someone when you are recording him/her. 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 your 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 developed by the instructor is the intellectual property of the instructor and cannot be shared publicly without his/her 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."

Academic integrity

Your commitment, as a student, to learning is evidenced by your enrollment at San Jose State University. The University Academic Integrity Policy S07-2 at http://www.sjsu.edu/senate/docs/S07-2.pdf 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. The Student Conduct and Ethical Development website is available at http://www.sjsu.edu/studentconduct/.

Campus Policy in Compliance with the American Disabilities Act

If you need course adaptations or accommodations because of a disability, or if you need to make special arrangements in case the building must be evacuated, please make an appointment with me as soon as possible, or see me during office hours. Presidential Directive 97-03 at http://www.sjsu.edu/president/docs/directives/PD_1997-03.pdf requires that students with disabilities requesting accommodations must register with the Accessible Education Center (AEC) at http://www.sjsu.edu/aec to establish a record of their disability.

Accommodation to Students' Religious Holidays

San José State University shall provide accommodation on any graded class work or activities for students wishing to observe religious holidays when such observances require students to be absent from class. It is the responsibility of the student to inform the instructor, in writing, about such holidays before the add deadline at the start of each semester. If such holidays occur before the add deadline, the student must notify the instructor, in writing, at least three days before the date that he/she will be absent. It is the responsibility of the instructor to make every reasonable effort to honor the student request without penalty, and of the student to make up the work missed. See University Policy S14-7 at http://www.sjsu.edu/senate/docs/S14-7.pdf.

CS 151 Object-Oriented Design, Spring 2017, Course Schedule

Schedule is subject to change with fair notice. Notice will be made available with Canvas notifications and announced in the class.

Week	Lecture#	Date	Topics	Comment / Deadline
1	1	01/28/17	Introduction, Java Refresher	
2	2	02/04/17	Object-Oriented Design Process	
3	3	02/11/17	Object-Oriented Design Process	
4	4	02/18/17	Guidelines for Class Design	Assignment #1
5	5	02/25/17	Guidelines for Class Design	
6	6	03/04/17	Interface Types and Polymorphism	
7	7	03/11/17	Patterns and GUI Programming	Assignment #2
8	8	03/18/17	Patterns and GUI Programming	
9	9	03/25/17	Inheritance and Abstract Classes	
10	10	04/01/17	Inheritance and Abstract Classes	Assignment #3
11	11	04/08/17	The Java Object Model	
12	12	04/15/17	The Java Object Model	
13	13	04/22/17	Multithreading	Assignment #4
14	14	04/29/17	Multithreading	
15	15	05/06/17	More Design Patterns	
16	16	05/13/17	Class Presentation	Final Project Due
17	17	05/20/17	Final Exam	Time: Place: 10:00 AM – 12:45 PM, MH 422