San José State University
College of Science/Department of Computer Science
CS46B, Object-Oriented Design, Section 1, Spring, 2019

Course and Contact Information

Instructor:	Nada Attar
Office Location:	MH 218
Telephone:	(408) 924-5108
Email:	nada.attar@sjsu.edu
Office Hours:	M 1:00-2:00pm | W 12:00-1:00pm
Class Days/Time:	M/W3:00-4:15am
Classroom:	Washington Square Hall 109
SI Leader(s)	Joseph Nguyen
SI Leader email (CC'd)	joseph.nguyen@sjsu.edu

Course Format

Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on my faculty web page at http://www.sjsu.edu/people/firstname.lastname and/or on Canvas Learning Management System course login website at http://sjsu.instructure.com. You are responsible for regularly checking with the messaging system through MySJSU at http://my.sjsu.edu (or other communication system as indicated by the instructor) to learn of any updates.

Prerequisite

Knowledge of Java equivalent to that obtained by completing CS 046A (in Java) or CS 049J with grade of C-or better. Eligibility for Math 030 or Math 030P, or instructor consent. Math remediation completed or a post baccalaureate. Pre/Co-requisite: Math 42. BRIN G HARDCOPY PROOF OF PRE-AND CO-REQUISITES TO 1st LECTURE

Course Description

Stacks and queues, recursion, lists, dynamic arrays, binary search trees. Iteration over collections. Hashing. Searching, elementary sorting. Big-O notation. Standard and custom collection classes.

Course Learning Outcomes (CLO)

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

1. Use and work with basic structures such as linked lists, stacks, queues, binary search trees, and iterators.
2. Implement Java classes that embody data structures.
3. Use pre-existing implementations such as the Java Collections framework.
4. Make relative estimates of the running times of alternative algorithms using big-O analysis.

5. Formulate and test for pre-and post-conditions.
6. Distinguish between different types of program defect, and understand how testing and debugging are used to correct them.
7. Implement simple sorting algorithms such as Insertion Sort and Selection Sort.
8. Implement the Sequential Search and Binary Search algorithms.
9. Implement simple recursive algorithms such as binary tree traversal.
10. Work competently with commonly used tools for software development.
11. Create custom data structures when appropriate pre-existing classes are not available.

Required Texts/Readings

Textbook

1. Preferred: Big Java 6e ENGAGE Custom Interactive Text by Cay S.Horstmann, ISBN: 9781119290223
2. Recommended: Big Java Early Objects 5th Edition by Cay S. Horstmann, ISBN 978-1-118-60771-8

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 S12-3at http://www.sjsu.edu/senate/docs/S12-3.pdf.

Homework

Homework assignments will be individual, regularly assigned, will include written problem assignments, and perhaps some online exercises. Solutions will be not posted. The homework is a tool for you to learn the material and prepare you for the exams.

There will be approximately 8 substantial programming assignments. No late homework will be accepted for any reason. The lowest homework grade will be dropped.

Midterm exams:

There will be two written Midterm exams during the semester.

Final Examination:

Section 1: Day: Monday, May 20 Time: 12:15-2:30pm
One written final cumulative exam.
The exams will contain multiple choice questions, short answer questions and questions that require pseudocode and/or computations. Makeup final exams will be only be given in cases of verifiable emergencies or, if the instructor is notified at least 3 weeks before the last class meeting, to students with at least 2 other finals in a 24-hour period.

Lab

All students must concurrently enroll in a lab section. The first lab meeting will be on Friday Feb 1st.
Labs are led by student TAs. All labs involve programming in Java, using the Eclipse IDE; try to install Eclipse before your first lab meeting.
Lab Rules:

1. Lab work must be done during lab sessions.
2. You may miss up to 2 scheduled lab meetings. If you miss more than 2 labs for any reason except for documented medical emergency, you will fail the course.
3. You cannot make up a missed lab.
4. Bring your laptop to each lab.

5. Work in pairs, unless there are an odd number of students, in which case 1 team will have 3 students. Teams will be formed at the first lab meeting; you choose your own lab partner. If your lab partner is absent, your lab instructor will form a team for you for that meeting.
6. A lab report is due at the end of each lab meeting. Reports will be graded from 0 to 4.
7. There are 2 roles, "Driver" and "Scribe", in each team. You and your lab partner will switch roles week to week. The driver runs Eclipse and submits a simple lab report. The scribe writes a more detailed lab report. Lab assignments clearly state what each report should contain.
8. If you and your partner are stuck, ask your lab instructor. Don't expect your instructor to give you answers; expect to be given ideas about how to get un-stuck.

Grading Information

Your grade for the course will be based on the following components:

- Mid Term Exams - 30 %
- Final Exam - 20 %
- Homework - 40 %
- Discretion - 5 %
- Lab reports - 5 %

Discretion includes participation in classes and answering forum posts on Piazza.

Midterm Exams are closed book; final exam is comprehensive. No extra point options. No make-ups exams except in case of verifiable emergency circumstances

Determination of Grades

The following shows the grading scale to be used to determine the letter grade:

Percentage	**Grade**
95 and above	A+
92-94	A
90 - 91	A-
87 - 89	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-
59 and below	F

Classroom Protocol

Attendance is highly recommended. Please avoid disturbing the class: turn-off cell phones (or put them on vibrate mode), no text messaging in the class or the exams, no taking pictures and video, avoid coming late.

You may not publicly share or upload material for this course such as exam questions, lecture notes, or solutions without my consent.

University Policies (Required)

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/" Make sure to review these policies and resources.

CS151, Object-Oriented Design, Section 1, Spring, 2019

Course Schedule

Week	Date	Topics, Readings, Assignments, Deadlines
1	M 1/28	Introduction & Course Mechanic
1	W 1/30	Introduction to data structures, Inheritance
2	M 2/04	Polymorphism
2	W 2/06	Inner Classes, Interfaces
3	M 2/11	Equality and comparison
3	W 2/13	Sets
4	M 2/18	Exceptions and Assertions
4	W 2/20	I/O
5	M 2/25	I/O (continue)
5	W 2/27	Review
6	M 3/04	**Midterm I**
6	W 3/06	Midterm solution & Recursion
7	M 3/11	Recursion & backtracking (continue)
7	W 3/13	Sorting & searching
8	M 3/18	Sorting & searching: algorithm complexity, big-O notation
8	W 3/20	The collections framework
9	M 3/25	Hash Tables
9	W 3/27	Trees
10	M 4/01	**Spring Break**
10	W 4/03	
11	M 4/08	Trees (continue)
11	W 4/10	Review
12	M 04/15	**Midterm II**
12	W 4/17	Midterm solution & Custom Collections
13	M 4/22	Custom Collections (continue)
13	W 4/24	Binary Search Trees
14	M 4/29	Problem solving & Tree algorithms
14	W 5/01	General graphs
15	M 5/06	General graphs
15	W 5/08	General graphs
16	M 5/13	Memory
16	W 5/15	Review
Final Exam	M 5/20	Time 12:15-2:30pm