CS250 Intro to Software Systems

Fall 2023

# Course Information

Class Days / Times: Thursday @ 7 pm

Class Location: GMCS, Rm 309

Mode: face-to-face lecture

Platform: Canvas

Instructor: Dr. Gus Hanna

Email: [gfhanna@sdsu.edu](mailto:gfhanna@sdsu.edu)

Zoom: sdsu.zoom.us/my/

Office hours: By appointment

I will attempt to respond to any email within 24 hours. Please include “[CS250]” in the subject line of any email related to class.

# [LAND ACKNOWLEDGMENT](https://sacd.sdsu.edu/diversity-resources/_pdfs/land-acknowledgement.pdf)

# ESSENTIAL STUDENT INFORMATION

For essential information about student academic success, please see the [SDSU Student Academic Success Handbook](https://docs.google.com/document/d/1rXNpNGs1K7nIxcS73o6R-fxZqPIWQwS9gHD7XpIqjhM/edit).

* SDSU provides disability-related accommodations via the Student Ability Success Center (sascinfo@sdsu.edu | [sdsu.edu/sasc](http://sdsu.edu/sasc)). Please allow 10-14 business days for this process.
* Class rosters are provided to the instructor with the student's legal name. Please let me know if you would prefer an alternate name and/or gender pronoun.

# COURSE MATERIALS

All course materials will be freely provided by the instructor on the Canvas course website. This includes readings, reference material, and lecture slides.

# COURSE DESIGN: MAJOR ASSIGNMENTS AND ASSESSMENTS

## Quizzes

## Quizzes will be announced in class and given via Canvas. The questions are based on the provided lecture content and associated readings.

## Quizzes are to be completed individually and answers should not be publicly posted. The goal of these quizzes is to direct your focus on the main concepts, ideas, and details presented. Students are encouraged to review the material and resources before/during/after completing these quizzes.

## Assignments

* Assignments will be presented in class and submitted via Canvas. Grading rubrics for each assignment will also be provided on Canvas.
* Group assignments are to be completed as a team with groups specified by the instructor, and only one submission per group is required. Groups will be assigned once, and remain the same for the entire semester.

## Final Report

* A Final Report will be required in lieu of a final exam. Students will compile a project report with their group and submit via Canvas.
* Each student will also be required to individually peer-review another group’s project report. Peer reviews will be assigned and completed via Canvas.
* Evaluation will be performed on both a group and individual basis.

## Participation

* In-class activities will be conducted throughout the semester. Attendance and participation in in-class activities will be reflected in this portion of the grade.

# COURSE SCHEDULE

**Table 2 COURSE SCHEDULE**

| Date | Activity | Assignment |
| --- | --- | --- |
| Week 1 | Introduction/Overview |  |
| Week 2 | Life-cycle Models |  |
| Week 3 | Requirements Engineering |  |
| Week 4 | Use Cases |  |
| Week 5 | Software Architecture | Asst 1: Requirements Specification |
| Week 6 | Class Diagrams |  |
| Week 7 | Verification and Validation | Asst 2: Design |
| Week 8 | Testing |  |
| Week 9 | Data Management | Asst 3: Test Plan |
| Week 10 | Networking |  |
| Week 11 | Security | Asst 4: Architecture 2.0 |
| Week 12 | Ethics |  |
| Week 13 | Project Review |  |
| Week 14 | Thanksgiving Recess |  |
| Week 15 | Project Group Work | Asst 5: Ethics and Communication  Final Project Report |
| Finals Week | Project Submission | Final Project Peer Review, Final Group Evaluation |

# GRADING POLICIES

Final grades in the course are determined based on the total weighted score. Letter grade cutoffs will be determined based on overall student performance - never above the standard letter grade distributions (93+ A, 90-92.99 A-, 87-89.99 B+, etc). Cutoffs are determined based on the student performance within the course offering, not based on percentage or number of students per letter grade. **This means if everyone does very well, everyone will get an A!**

Note that the course grading scheme may be modified during the course if an issue with overall lack of participation, widespread academic dishonesty, or disruptive extraneous circumstances emerge.

Grading breakdown:

* Assignments: 50% (10% each)
* Final Report: 30%
* Participation: 15%
* Quizzes: 5%

## Late Work

Following the true spirit of software deliverables, late assignments will be accepted up to 5 days after the deadline. For late submissions, 10% will be deducted from the assignment grade for each day after the deadline. Maintaining the spirit of software deliverables, the penalty applies to the entire group for group assignments, even if each individual has a deliverable.

## Contesting Grades

An assignment’s grade must be contested within two weeks of its due date (even if submitted late). To contest a grade, submit a written justification via email to the TA detailing specifically why you think your grade should be adjusted, referring to the rubric.

## Academic Dishonesty

[Don’t cheat.](https://sacd.sdsu.edu/student-rights/academic-dishonesty/cheating-and-plagiarism)

## Accommodations

If you are a student with a disability and believe you will need accommodations for this class, it is your responsibility to contact the Student Ability Success Center at (619) 594-6473. To avoid any delay in the receipt of your accommodations, you should contact the Student Ability Success Center as soon as possible. Please note that accommodations are not retroactive and that I cannot provide accommodations based upon disability until I have received an accommodation letter from the Student Ability Success Center. Your cooperation is appreciated.

# STUDENT LEARNING OUTCOMES

Degree Student Outcomes Supported:

* SO 3: Graduates of the program will have the ability to communicate effectively in a variety of professional contexts.
* SO 4: Graduates of the program will have the ability to recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.

Course Learning Outcomes:

* CLO 1: Identify the functions of various software system components.
* CLO 2: Identify alternative approaches for various software system functions and integrate them into a system model
* CLO 3: Design a solution model based on a set of project requirements.
* CLO 4: Function effectively in teams to accomplish model designs and analyses.