Objects and Design

Last Updated: Wed, 01/07/2026

Course prefix: CS

Course number: 2340

Section: A

CRN (you may add up to five):
24970

Instructor First Name: Pedro

Instructor Last Name: Feijoo Garcia

Semester: Spring

Academic year: 2026

Course description:

Object-oriented programming methods for dealing with large programs. Focus on quality processes, effective debugging techniques, and testing to ensure a quality product. CS2340 takes students who know an object-oriented language and focuses on getting them to use that language in a true object- oriented style. The course achieves this goal by introducing a design methodology and notation, and covering standard principles and practice in design.

Course learning outcomes:

Core Outcomes: The primary outcomes are:

1. Improve existing object-oriented programming skills.

2. Complete a team-based large-scale programming project as a team.

3. Use industry tools and practices to implement a large-scale project.

4. Conduct object-oriented analysis and design and document it with

standard techniques (UML).

• General Outcomes:

1. (Movement - Synthesis) Improve existing programming skills by developing much larger and more complex programs than in previous classes.

2. (Accomplishment - Synthesis) Given a requirements list, complete a team-based large-scale programming project that implements thoserequirements. The project will require at least 3000 lines of code and multiple compilation modules (or equivalent jars) to complete.

3. (Experience - Analysis) Reflect on the difficulties of team membership and the challenges of developing software in a team environment.

4. (Competency - Application) Demonstrate the ability to use a version control system such as Git to manage team code.

5. (Competency - Application) Demonstrate the ability to use standard tools to help with large-scale projects such as commercial quality development environments (e.g., PyCharm, IntelliJ, or Eclipse).

6. (Movement - Synthesis) Improve object-oriented development skills by learning to think about objects when faced with a design problem. This is evidenced by the minimal use of class methods and data and the proper use of abstraction, information hiding, and encapsulation.

7. (Competency - Synthesis) Given a specification of requirements, analyze those requirements using domain models, use cases, and robustness diagrams. Select appropriate candidate objects representing the problem domain.

8. (Competency - Synthesis) Given a set of use cases and domain models representing a customer problem, design an object-oriented solution

and document that solution using the Unified Modeling Language (UML).

9. (Competency - Analysis) Apply standard design principles and patterns to a problem specification. Analyze a proposed design to determine its

compliance with the standard principles (e.g., open-closed, dependency inversion, Law of Demeter) and make corrections as necessary.

10. (Achievement - Synthesis) Given a problem specification, design, document, and implement an object-oriented solution as a development team.

11. (Competency - Analysis) Demonstrate the ability to derive open and enclosed tests from code or specifications. Document those tests in a basic test plan and implement those tests using an automated test environment such as JUnit.

Required course materials:

1. Correa, D., & Lim, G. (2024). Django 5 for the impatient: Learn the core concepts of Django to develop Python web applications.
2. Ko, A. J. (2025). Cooperative software development https://faculty.washington.edu/ajko/books/cooperative-software-development

(Retrieved July 28, 2025)

3. Ko, A. J. (2025). Design methods. https://faculty.washington.edu/ajko/books/design-methods/ (Retrieved July 28, 2025)
4. Rozanski, N., & Woods, E. (2012). Software systems architecture: Working with stakeholders using viewpoints and perspectives (2nd ed.). Addison-Wesley.
5. Freeman, E., Robson, E., Bates, B., & Sierra, K. (2004). Head First Design Patterns: A Brain-Friendly Guide. " O'Reilly Media, Inc."

Grading policy:

Letter grades:

Letter grades are assigned according to the following convention:

- A = [90, 100) points.

- B = [80, 90) points.

- C = [70, 80) points.

- D = [60, 70) points.

- F = [0, 60) points.

Keep in mind that point values (fractional or otherwise) are not rounded to the next grade level. For example, 89.99 will be reported as "B." 90.00 (or higher) will be reported as "A." There are no exceptions.

- Regrade Requests:

Once graded assignments and/or assessments are returned, there is a five-days window during which you can revise and request grade updates. The regrade period begins when the grades are released on Canvas. Late regrade requests will not be considered. The regrade period (i.e., five-days window) considers weekend days and holidays.

It is the student's responsibility to keep track of their performance in the course. The instructor's grades will be assumed to be accurate unless the student can prove otherwise. Always keep a digital copy of ALL work turned in to your instructional team. Any student wishing for a regrade must submit a written document indicating the specific section the student is requesting a regrade of and a complete explanation (rationale) of why the student considers why they deserve a different grade. Please consider:

- Verbal regrade requests will not be accepted.

- The instructional team reserves the right to regrade the entire submission and not just the specific portion in question. Regrade requests mean a potential update to the entire or partial submission. That is, the student's grade can be raised or lowered by the regrade request.

Attendance policy:

Attendance to class is mandatory, and your participation is required. Engagement with the course is required to be successful in the course. Lectures will cover material that may not be available on Canvas. It is the student’s responsibility to catch up with fellow classmates on notes and topics missed in the case of an absence. Any participation activities will be excused for absences that are verified by the Dean ofStudents Office (DoS). If a student misses class and feels it is excused, they may submit their documentation to the Dean of Students Office. All absences must be approved by the Dean of Students Office.

You may notify me (your instructor) about your absence, but please do not send me your documentation directly. I will only excuse the activity when I receive notification from the DoS Office. Excusing any activities may not be done on Canvas until the end of the semester to provide the student with a "worst-case scenario;" the student is encouraged to compute their grade offline outside of Canvas based on the weights outlined in Canvas.

Participation in all team-based assignments is required and expected, and your grades will be affected based on your contribution and response to your assigned tasks in your teams.

Academic honesty/integrity statement:

Students are expected to maintain the highest standards of academic integrity. All work submitted must be original and properly cited. Plagiarism, cheating, or any form of academic dishonesty will result in immediate consequences as outlined in the university's academic integrity policy. Using third-party libraries or tools that are not explicitly mentioned requires the permission of the instructional team.

The use of copyrighted or offensive material in your projects is prohibited and will be sanctioned through the Office of Student Integrity (OSI).