CS 3331: Advanced Object-Oriented Programming
Fall 2023

CRN: 13588
Lecture: TR 3:00 PM - 4:20 PM in CCSB G.0208
Website: Blackboard
Instructor: Yoonsik Cheon (x-8028, ycheon@utep.edu); office hours: TR 4:30 PM - 5:20 PM in CCSB 3.0606
TA: TBA
Prerequisite: CS 2302 with a grade of C or better

Description

Taken from the University Catalog: "An in-depth exposure to the object-oriented programming paradigm, which builds upon programming experience gained in lower-level computer science classes. Emphasis on programming in an object-oriented language with which students are already familiar, and on requirements, testing, code reading, and comprehension."

Objectives

This course aims to empower students with advanced design and programming techniques within the realm of object-oriented programming paradigms (please refer to the detailed learning outcomes on page 5). To achieve this, the course has set forth specific objectives:

- Develop a profound comprehension of object-oriented design concepts and principles.
- Cultivate the essential skills required for crafting high-quality object-oriented programs.
- Attain proficiency in utilizing object-oriented design notations and supporting tools, such as UML, to model solutions for problems and software systems effectively.
- Attain a high level of proficiency in navigating and operating within object-oriented programming environments.

Textbooks

This course relies on two required textbooks, both of which are conveniently accessible for free through the UTEP Library:

- David Parsons. *Foundational Java: Key Elements and Practical Programming*. Second edition, Springer, 2021.
- Matt Weisfeld. *The Object-Oriented Thought Process.* Fifth edition, Addison-Wesley Professional, 2019.

In addition to the aforementioned essential textbooks, we recommend the following supplementary reading materials to enhance your understanding:

- Cay S. Horstmann. *Core Java for the Impatient*, 3rd edition, Addison-Wesley, 2022. *
- Martina Seidl, et al. *UML@Classroom: An Introduction to Object-Oriented Modeling.* Springer, 2015. *
- Scott W. Ambler. *The Elements of UML 2.0 Style*. Cambridge University Press, 2005.
- Allan Vermeulen, et al. *The Elements of Java Style*. Cambridge University Press, 2000.

* Please note that the books marked with an asterisk (*) have free e-book versions accessible through the UTEP Library.

Exams

The course will include a mid-term exam and a final exam. The mid-term exam is scheduled during a regular class session and will have a duration of 80 minutes. The final exam is set for the date specified by the university. A makeup exam will be considered only under exceptional or unavoidable circumstances, such as severe illness or a commitment

to present at a conference. If you find yourself in a situation that may require a makeup exam, it is imperative to promptly inform the course staff. For those planning to attend a conference for a paper presentation, exam arrangements must be made in advance. In all situations, submission of an official document explaining your circumstances is mandatory prior to scheduling a makeup exam.

Homework

Throughout the course, you can expect to have multiple homework assignments. These assignments will be communicated either during class sessions or via the Blackboard platform. Should you happen to miss a class, it remains your responsibility to ascertain the content you may have missed.

Most of the homework assignments will involve Java programming. Unless otherwise specified, each assignment should be completed independently. While discussions about the assignments in broad terms with your peers are permitted, the actual design, coding, and testing of your solutions should be executed individually. Plagiarism, including copying and pasting from the internet, is strictly prohibited; all work should be expressed in your own words.

Should you require assistance, don't hesitate to approach either the instructor or the teaching assistant (TA). Submission of your completed assignments is to be carried out exclusively through the Blackboard platform. Please note that unless prearranged or exceptional circumstances dictate otherwise, late submissions will not be accepted.

Grading

In this course, your grade is determined independently of your peers' grades. We do not employ a curved grading system; all students have the potential to achieve an A grade. The primary objective of grading is not to rank individuals but rather to maintain a standard of excellence and provide constructive feedback.

Your final letter grade will be calculated based on a combination of various components, including lessons (which encompass readings, quizzes, exercises, and more), homework assignments, and exams. The following approximate percentages illustrate the weight assigned to each activity:

Activities	Percent (%)
Lessons (readings, quizzes, exercises, etc.)	30
Homework assignments	35
Exams	35

Additionally, there exists an opportunity to earn up to 5% bonus points for class attendance and active participation. To secure this bonus, your engagement should involve punctual attendance, constructive contributions to class discussions, and demonstrated preparedness, such as asking and answering questions that showcase your engagement with the material. Completing in-class assignments and activities promptly is also crucial. We will gauge your participation through various means, including Blackboard tracking tools, discussions, blogs, chat sessions, and group work.

It is essential to adhere to due dates diligently. Make-up assignments or late submissions will only be considered with a compelling and legitimate reason, subject to the instructor's approval. For the submission of work, please follow the guidelines provided unless instructed otherwise. Submissions must be made electronically through Blackboard. Ensure your work is submitted by the stipulated due date, or seek special permission from the instructor before the deadline. Generally, extensions beyond the next assignment will be granted only under exceptional circumstances.

Final letter grades will be calculated based on the percentage of total points earned. The following table presents the nominal percentage-score-to-letter-grade conversion:

Letter grade	Percent (%)	Performance
A	90-100	Excellent
B	80-89	Good
C	70-79	Average
D	60-69	Poor

F	0-59	Failing

The instructor retains the right to make downward adjustments to these criteria based on the overall performance of the class. For instance, a score of 88% or higher may be deemed deserving of an A grade.

Your commitment and dedication to your studies will greatly influence your performance in this course. Please feel free to reach out to the instructor for any clarifications or assistance you may require.

Attendance

Regular class attendance is mandatory, as consistent attendance significantly contributes to your success in the course. Attending classes regularly is essential for achieving optimal results. Please note that the instructor retains the right to address unexcused absences with appropriate measures. For instance, your final grade may be subject to a reduction of one point for each unexcused absence beyond three instances. The guidelines provided below are extracted from the 2023-2024 Catalog:

"The student is expected to attend all classes and laboratory sessions and attendance is mandatory for all freshman-level courses. It is the responsibility of the student to inform each instructor of extended absences. When, in the judgment of the instructor, a student has been absent to such a degree as to impair his or her status relative to credit for the course, the instructor can drop the student from the class with a grade of W before the course drop deadline and with a grade of F after the course drop deadline."

Standards of Conduct

Your behavior in this course is expected to align with professional and courteous standards, as outlined in the Handbook of Operating Procedures: Student Conduct and Discipline. It is imperative that all graded assignments (quizzes, exercises, homework, exams) are completed independently and reflect unmistakably your individual effort, even though general discussions with peers are permissible. Material borrowed or copied from any external source, be it individuals, books, or online resources, cannot be presented as your original work.

> "Plagiarism means the appropriation, buying, receiving as a gift, or obtaining by any means another's work and the unacknowledged submission or incorporation of it in one's academic work offered for credit or using work in a paper or assignment for which the student had received credit in another course without direct permission of all involved instructors. Plagiarism is a serious violation of university policy and will not be tolerated."

It is crucial to note that instances of suspected plagiarism will be reported to the Office of Student Conduct and Conflict Resolution (OSCCR) in accordance with university requirements. We are committed to upholding academic integrity and fostering an environment of learning based on honesty and originality. Your adherence to these principles is expected and appreciated.

Accommodations

If you have a disability that requires classroom accommodation, please reach out to the Center for Accommodations and Support Services (CASS) at 747-5148 or via email at cass@utep.edu. Their office is situated in UTEP Union East, Room 106. For more details, kindly visit the CASS website at http://www.sa.utep.edu/cass. Your comfort and accessibility are of utmost importance, and CASS is here to assist you in ensuring a conducive learning experience.

Schedule

The table below presents a planned schedule for the course. For the most current schedule, consult the course website.

Dates		Topics	Readings			Assignments
			[FJ]	[OOTP]	[UC]	
Week 1	Aug. 29, 31	Intro to CS 3331 OO concepts		Ch 1-2		
Week 2	Sep. 5, 7	OO modeling with UML OO application frameworks			Ch 3	HW1: UML
Week 3	Sep. 12, 14	Java 2D graphics				
Week 4	Sep. 19, 21	Association, aggregation, and composition	Ch 7	Ch 7, 9	Ch 4	HW2: App
Week 5	Sep. 26, 28	Inheritance, polymorphism, and interface	Ch 8	Ch 7, 8	Ch 4	
Week 6	Oct. 3, 5	Class design API documents (Javadoc)	Ch 5-6 Se 5.5, 6.5	Ch 5		
Week 7	Oct. 10, 12	Exception handling Unit testing (JUnit)	Ch 9 Ch 10			HW3: JUnit
Week 8	Oct. 17, 19	GUI **Exam 1**	Ch 17	Ch 8		
Week 9	Oct. 24, 26	GUI	Ch 18 Ch 19			HW4: GUI
Week 10	Oct. 31, Nov. 2	GUI Design patterns		Ch 10		
Week 11	Nov. 7, 9	Design patterns				
Week 12	Nov. 14, 16	Network programming	Ch 20			HW5: Net
Week 13	Nov. 21, 23	Multithreading	Ch 10 of CJ			
Week 14	Nov. 28, 30	Collections and I/O	Ch 12-13			
Week 15	Dec. 5, 7	Design principles Programming work		Ch 11-12		
Week 16	Dec. 14	**Final** at 1:00 PM - 3:45 PM				

[FJ] David Parsons. *Foundational Java: Key Elements and Practical Programming*. Second edition, Springer, 2021.
[OOTP] Matt Weisfeld. *The Object-Oriented Thought Process*. Fifth edition, Addison-Wesley Professional, 2019.
[UC] Martina Seidl, et al. *UML@Classroom: An Introduction to Object-Oriented Modeling.* Springer, 2015.
[CJ] Cay S. Horstmann. *Core Java for the Impatient.* Third edition, Addison-Wesley, 2022.

Important Dates

August 28:	Classes begin
September 4:	Labor Day holiday – university closed
September 13:	Census Day
October 19:	Exam 1
November 3:	Course drop/withdrawal deadline
November 23-24:	Thanksgiving holiday - university closed
December 7:	Last day of classes
December 8:	Dead day
December 14:	Final on Thursday at 4:00 pm – 6:45 pm

CS 3331: Advanced Object-Oriented Programming
Learning Outcomes

Level 1: Knowledge and Comprehension

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level. Upon successful completion of this course, students will be able to:

a. Explain the differences between an object-oriented approach and a procedural approach.
b. Describe the difference between waterfall and agile software development.

Level 2: Application and Analysis

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able to:

a. Formulate use-case diagrams and scenarios to support understanding of user requirements.
b. Use object-oriented design notations, including UML class diagrams and state machine diagrams (optionally sequence diagrams) to model problem solutions.
c. Use basic object-oriented design patterns to structure solutions to software design problems.
d. Translate design features, such as classes and relationships, to implementations.
e. Use frameworks and library classes and methods, such as collections, GUI, multithreading, and networking, in problem solutions.

Level 3: Synthesis and Evaluation

Level 3 outcomes are those in which the students can apply the material in new situations. This is the highest level of mastery. Upon successful completion of this course, students will be able to:

a. Design and implement software employing the principles of modularity, encapsulation, information hiding, abstraction, and polymorphism.
b. Design, implement, and use classes and objects by following coding conventions, guidelines, styles, and standards.
c. Design, and implement exception handling (including user-defined exceptions) and high-order functions.
d. Evaluate existing classes and software for the purposes of extension through inheritance.
e. Create API documents for classes, fields, and methods.
f. Design and implement test suites for automated unit testing.
g. Re-factor existing source code to improve its design or efficiency.