

CS 3331: Advanced Object-Oriented Programming Spring 2023

CRN: 24416

Lecture: MW 3:00 PM - 4:20 PM in CRBL C305

Website: Blackboard

Instructor: Yoonsik Cheon (x-8028, ycheon@utep.edu); office hours: MW 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 will equip students with advanced design and programming techniques in object-oriented programming paradigms (refer to the learning outcomes on page 5). To this end, specific objectives are:

- To understand object-oriented design concepts and principles,
- To acquire skills needed for developing high-quality object-oriented programs,
- To be able to use object-oriented design notations and support tools such as UML for modeling problem solutions and software systems, and
- To be proficient in object-oriented programming environments.

Textbooks

There are two required textbooks, and both books are 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 above-required textbooks, the following books are recommended for supplementary reading:

- 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.

*A free e-book is available through the UTEP Library.

Exams

There will be one mid-term exam and the final exam. The mid-term exam will take place during the regular class session and will be 80 minutes long, and the final exam will take place on the date specified by the university. We grant a makeup exam only in an unusual or unavoidable circumstance, e.g., an incapacitating illness or a presentation at a conference. If you are in this circumstance that may warrant a makeup exam, you must notify the course staff as early as possible. If you plan to attend a conference to present a paper, you must arrange your exam *in advance*. In all circumstances, you must provide an official document explaining your situation before your make-up exam.

Homework

There will be several homework assignments, and all will be announced in class or through Blackboard. If you miss a class, you are responsible for finding out what you missed. Most homework requires Java programming. You must do all your homework individually unless specified otherwise; refer to the handouts for possible pair work. It means that while you may discuss the assignment in general terms with your classmates, you should design, write, and test your code alone. Also, do not copy and paste text or code from the Internet; write them in your own words. If you need help, see the instructor or the TA. You must submit all your work through Blackboard, and no late submission will be accepted unless arrangements are made in advance or unless unusual circumstances warrant an exception.

Grading

Your grade is independent of everyone else's grade. We do not grade on a curve; everyone can earn an A. The purpose of grading is not to rank you but to uphold a standard of quality and give feedback. We will calculate your final letter grade using a combination of lessons (quizzes and exercises), homework assignments, and exams. Shown below are the approximate percentages:

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

There are also up to 5% bonus points for class attendance and participation. To earn this bonus, you must arrive at lectures on time and participate in class discussions in a constructive and prepared manner, e.g., by asking or answering questions that demonstrate that you have read and attempted to understand the material. You should also complete classwork and activities on time. We will monitor, track, and score your participation in the course partly using Blackboard tracking tools, discussions, blogs, chat sessions, and group work.

Be sure to pay close attention to due dates -- there will be no makeup assignments or late work accepted without a serious and compelling reason and the instructor's approval. You should submit all your work for this course electronically through Blackboard unless otherwise instructed. You must submit your work by the due date or obtain special permission from the instructor before the due date. Except under extreme circumstances, you won't get an extension beyond the next assignment.

We will calculate your final letter grades based on the percentage of your total points earned. Below is 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 reserves the right to adjust these criteria downward by considering overall class performance, e.g., 88% or higher represents an A.

Attendance

Class attendance is required; you should understand that your success in the course will improve by attending classes regularly. The instructor reserves the right to penalize unexcused absences, e.g., your final grade may be lowered by one point for each unexcused absence above three. The following is excerpted from the 2022-2023 Catalog.

“The student is expected to attend all classes and laboratory sessions and attendance are 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

You should conduct yourself professionally and courteously as described by the Handbook of Operating Procedures: Student Conduct and Discipline. All graded work (quizzes, exercises, homework, exams) is to be completed independently and should be unmistakably your work, although you may discuss your work with others in a general way. You may not represent the material transcribed or copied from another source, including persons, books, or Web pages, as your 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.”

As required by the university, we will report all cases of suspected plagiarism to the Office of Student Conduct and Conflict Resolution (OSCCR) for further review.

Disabilities

If you have a disability and need classroom accommodations, contact the Center for Accommodations and Support Services (CASS) at 747-5148, or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at <http://www.sa.utep.edu/cass>.

Schedule

Dates		Topics	Readings			Assignments
			[FJ]	[OOTP]	[UC]	
Week 1	Jan. 18	Intro to CS 3331				
Week 2	Jan. 23, 25	OO concepts OO modeling with UML		Ch 1-2	Ch 3	HW1: UML
Week 3	Jan. 30, Feb. 1	OO application frameworks Java 2D graphics				
Week 4	Feb. 6, 8	Association, aggregation, and composition	Ch 7	Ch 7, 9	Ch 4	HW2: App
Week 5	Feb. 13, 15	Inheritance, polymorphism, and interface	Ch 8	Ch 7, 8	Ch 4	
Week 6	Feb. 20, 22	Class design API documents (Javadoc)	Ch 5-6 Se 5.5, 6.5	Ch 5		
Week 7	Feb. 27, Mar. 1	Exception handling Unit testing (JUnit)	Ch 9 Ch 10			HW3: JUnit
Week 8	Mar. 6, 8	GUI Exam 1	Ch 17	Ch 8		
Week 9	Mar. 13, 15	Spring break (no classes)				
Week 10	Mar. 20, 22	GUI	Ch 18 Ch 19			HW4: GUI
Week 11	Mar. 27, 29	GUI Design patterns		Ch 10		
Week 12	Apr. 3, 5	Design patterns				
Week 13	Apr. 10, 12	Network programming	Ch 20			HW5: Net
Week 14	Apr. 17, 19	Multithreading	Ch 10 of CJ			
Week 15	Apr. 24, 26	Collections and I/O	Ch 12-13			
Week 16	May 1, 3	Design principles		Ch 11-12		
Week 17	May 8	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

January 16: Dr. Martin Luther King, Jr. holiday – university closed

January 17: Classes begin

February 1: Census Day

March 11: Exam 1

March 13-17: Spring break

March 30: Course drop/withdrawal deadline

March 31: Cesar Chavez holiday - no classes

April 5: Course drop/withdrawal deadline

April 7: Spring study day

May 4: Last day of classes

May 5: Dead day

May 8: Final on Monday at 1:00 PM - 3:45 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.