

CS 3331: Advanced Object-Oriented Programming

Fall 2009

CRN: 14917

Lecture: MW 1:30 pm - 2:50 pm in CS 322

Website: <http://www.cs.utep.edu/cheon/cs3331>

Instructor: Yoonsik Cheon (x-8028, ycheon@utep.edu); office hours: MW 10:30-11:50 am in CS 202B

TA: Cesar Yeep (x-8949, ceyeep@miners.utep.edu); office hours: TR 12:30-3:00 pm in CS 128

Prerequisite: CS 2402 with a grade of C or better

Course Goals

The goal of this course is to equip students with advanced design and programming techniques in the object-oriented programming paradigms. To this end, specific goals are:

- To increase students' knowledge of object-oriented design concepts,
- To teach the knowledge and skills needed to develop reusable, quality programs,
- To instruct students on the use of object-oriented design tools such as UML for modeling problem solutions and complex systems, and
- To increase students' proficiency in programming in object-oriented environments.

Learning Outcomes

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

- [Level 3: *Synthesis and evaluation*] Develop modular solutions to a given problem statement,
- [Level 3] Design and implement software employing the principles of encapsulation, information hiding, abstraction, and polymorphism,
- [Level 3] Design, implement, and use classes and methods in an object-oriented programming language, employing standard naming conventions and making appropriate use of advanced features such as inheritance, exception handling, I/O, references, and simple GUIs,
- [Level 3] Evaluate existing classes and software for the purposes of extension through inheritance,
- [Level 3] Use and create standard API documents to understand and document the use of classes and methods,
- [Level 3] Design and implement through test suites (unit testing),
- [Level 3] Refactor existing software to improve its design or efficiency,
- [Level 2: *Application and analysis*] Use object-oriented design tools such as UML class diagrams to model problem solutions and express inheritance, association, aggregation, and composition relationships among classes,
- [Level 2] Recognize and use basic object-oriented design patterns to structure solutions to problems,
- [Level 2] Implement association relationships and multiplicities,
- [Level 2] Use frameworks, classes, and methods from standard libraries in problem solutions,
- [Level 1: *Knowledge and comprehension*] Explain the fundamentals of software development including development process, quality of software systems, and challenges of software development, and
- [Level 1] Define or explain principles of modularity, encapsulation, information hiding, abstraction, and polymorphism.

Texts

The required course textbook is Xiaoping Jia's *Object-Oriented Software Development Using Java*, 2nd edition, Addison Wesley, 2003. The textbook is available at the UTEP bookstore, and you are expected to acquire a copy for your use in this course, as reading assignments will be taken from the textbook.

In addition to the required textbook, the following books are recommended as references:

- 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.

- Maurice Naftalin and Philip Wadler. *Java Generics*, Oreilly, 2007.

Assignments

There will be two kinds of assignments: individual assignments and group assignments. All assignments will be announced or handed out in class. If you miss a class, it is your responsibility to find out what you missed. A majority of assignment will require programming in Java. For non-programming assignments, **no late submissions will be accepted** unless arrangements have been made in advance or unless unusual circumstances warrant an exception. For programming assignments, late submissions will be accepted, but **programs turned in late will be penalized 10% for each day or partial day of lateness** for up to five days. After five days, no submission will be accepted unless other arrangements have been made in advance or unless unusual circumstances warrant an exception. All individual assignments, whether programming or not, are to be done individually. While you may discuss the assignment in general terms with others, your solutions should be composed, designed, written and tested by you alone. If you need help, consult the TA or the instructor.

Exams

There will be one mid-term exam and one final exam. The final exam will be comprehensive. The mid-term exam will take place during the regular class session and will be 60 minutes in length, and the final exam will take place on the date specified by the university.

Grading

Your semester grade will be based on a combination of individual assignments, group assignments, and exams. The approximate percentages are as follows:

Individual assignments:	35%
Group assignments:	35%
Exams:	30%

In addition, a bonus of up to 5% is available for lecture attendance and participation. To earn this bonus, you must arrive at lecture on time and participate in class discussion 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.

The nominal percentage-score-to-letter-grade conversion is as follows:

90% or higher is an A
80-89% is a B
70-79% is a C
60-69% is a D
below 60% is an F

I reserve the right to adjust these criteria downward, e.g., so that 88% or higher represents an A, based on overall class performance. The criteria will not be adjusted upward, however.

Attendance

Lecture attendance is not mandatory but is recommended. You should understand that your success in the course will improve greatly by attending class regularly. It is your responsibility to keep up to date with notes, assignments and exam.

Standards of Conduct

You are expected to conduct yourself in a professional and courteous manner, as prescribed by the UTEP Standards of Conduct. Graded work (homework, projects, exams) is to be completed independently and should be unmistakably your own work, although you may discuss your work with others in a general way. You may not represent as your own work material that is transcribed or copied from another source, including persons, books, or Web pages. Instructors are required to—and will—report academic dishonesty and any other violation of the Standards of Conduct to the Dean of Students.

Disabilities

If you feel that you may have a disability that requires accommodation, contact the Disabled Student Services Office at 747-5184, go to Room 106E Union, or email dss@utep.edu.

Schedule

The following table shows a planned schedule for the course. The schedule is subject to change, and an up-to-date schedule will be available from the course website.

Dates		Topics	Readings	Assignments
Week 1	Aug. 24	OO software development OO modeling with UML	Chap 1 & 2	
Week 2	Aug. 31	UML Documenting code (Javadoc)	Chap 2 Sec 6.1.2	UML diagrams
Week 3	Sep. 7	Applets Advanced Java features	Sec 3.3 Sec 4.4-4.7	Applet program
Week 4	Sep. 14	Unit testing (JUnit) Class and inheritance	Sec 6.4 Chap 5	JUnit
Week 5	Sep. 21	Class and inheritance Class design	Chap 5 Sec 6.1 & 6.3	
Week 6	Sep. 28	Design by contract (JML)	Sec 6.2 & handout	JML
Week 7	Oct. 5	Review for exam 1 Exam 1		
Week 8	Oct. 12	Design by abstraction	Chap 7	Project: requirements
Week 9	Oct. 19	Frameworks and GUI	Sec 8.1-8.4	GUI program
Week 10	Oct. 26	GUI, collections, and I/O	Sec 8.1-8.4	
Week 11	Nov. 2	Design patterns	Chap 10	Project: design
Week 12	Nov. 9	Design patterns	Chap 10	
Week 13	Nov. 16	Concurrent programming	Chap 11	Applet program
Week 14	Nov. 23	Network programming	Chap 12	
Week 15	Nov. 30	Project presentation		Project: report
Week 16	Dec. 9	Final at 4:00 pm – 6:45 pm		

Important Dates

August 24: Class begins
 September 7: Labor day (no class)
 September 9: Census day
 October 9: Exam 1
 October 30: Course drop deadline
 November 6: Exam 2
 November 26-27: Thanksgiving holidays (no class)
 December 4: Dead day (no class)
 December 9: Final on Wednesday at 4:00 pm – 6:45 pm