

# CS 3331: Advanced Object-Oriented Programming Spring 2019

CRN: 22041

Lecture: MW 9:00 - 10:20 am in CCSB G.0208

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

Instructor: Yoonsik Cheon (x-8028, ycheon@utep.edu); office hours: MW 1:30-3:00 pm in CCSB 3.0606

TA: Ubaldo Castro (ucastro@miners); office hours: TBA

Prerequisite: CS 2302 with a grade of C or better

## Course Objectives

This course is to equip students with advanced design and programming techniques in the object-oriented programming paradigms. 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 of 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.

## 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:

- 1a. Explain the differences between an object-oriented approach and a procedural approach.

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:

- 2a. Formulate use-case diagrams and scenarios to support understanding of user requirements.
- 2b. Use object-oriented design notations, including UML class diagrams and state machine diagrams (optionally sequence diagrams) to model problem solutions.
- 2c. Use basic object-oriented design patterns to structure solutions to software design problems.
- 2d. Translate design features, such as classes and relationships, to implementations.
- 2e. 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:

- 3a. Design and implement software employing the principles of modularity, encapsulation, information hiding, abstraction, and polymorphism.
- 3b. Design, implement, and use classes and methods that follow conventions and styles, and make appropriate use of advanced features such as inheritance, exception handling, and generics.
- 3c. Evaluate existing classes and software for the purposes of extension through inheritance.
- 3d. Create API documents for classes, fields and methods.
- 3e. Design and implement test suites for automated unit testing.
- 3f. Re-factor existing source code to improve its design or efficiency.

## Textbooks

The course textbook is Xiaoping Jia's *Object-Oriented Software Development Using Java*, 2<sup>nd</sup> edition, Addison Wesley, 2002. The textbook should be 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 for supplementary reading:

- Martina Seidl, et al., *UML@Classroom: An Introduction to Object-Oriented Modeling*, Springer, 2015. \*
- Cay S. Horstmann, *Core Java Volume I – Fundamentals*, 11<sup>th</sup> edition, Prentice Hall, 2018 \*  
(or Cay S. Horstmann, *Core Java SE 9 for the Impatient*, 2<sup>nd</sup> edition, Addison-Wesley, 2017\*).
- 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.

\*E-books are available to UTEP users through UTEP Library; use VPN from outside the UTEP domain .

## Homework

There will be several homework assignments, and most homework will require programming in Java. Some may be done in pairs or teams. For an individual assignment, while you may discuss the assignment in general terms with others, your solutions (programs) should be composed, designed, written and tested by yourself alone. If you need a help, consult with the TA or the instructor. All homework assignments will be announced or handed out in class; if you miss a class, it is your responsibility to find out what you missed. No late submission will be accepted.

## Exams

There will be one mid-term exam and the final exam. The final exam will be comprehensive. The mid-term exam will take place during the regular class session and will be 80 minutes in length, and the final exam will take place on the date specified by the university.

Make-up exams will be given only when you have unusual circumstances, such as incapacitating illness or presenting a research paper at a conference. If you believe that you have an unusual circumstance that warrants a make-up exam, notify us as soon as possible. If you will be attending a conference or other event, you must make arrangements for a make-up exam in advance. Under any circumstances, you may be required to provide official documentation before a make-up will be administered.

## Grading

Your grade is independent of anyone else's grade. We do not grade on a curve, and everyone can earn an A. The purpose of grading is not to rank you, but to uphold a standard of quality and to give you feedback. Your final letter grade will be calculated based on a combination of quizzes, in-class work, homework assignments, exams, and class participation. The approximate percentages are shown below:

Quizzes:	15%
Homework assignments:	50%
Exams:	35%

In addition, bonus points of up to 5% is available for class attendance and participation. To earn this bonus, you must arrive at classes 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.

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

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

### **Attendance**

Class attendance is required; you should understand that your success in the course will improve greatly 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 2017-2018 Undergraduate Catalog.

The student is expected to attend all classes and laboratory sessions. 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 are expected to conduct yourself in a professional and courteous manner, as prescribed by the Handbook of Operating Procedures: Student Conduct and Discipline. All 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. "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 own 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. All cases of suspected plagiarism will be reported to the Dean of Students for further review.

### **Disabilities**

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

## Schedule

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

Dates		Topics	Readings	Assignments
Week 1	Jan. 21, 23	Intro to CS 3331 OO software development	Chap 1	
Week 2	Jan. 28, 30	OO modeling with UML Class diagram	Sec 2.4-2.5 (UC Chap 3)* Sec 2.1-2.2 (UC Chap 4)	HW1
Week 3	Feb. 4, 6	Class diagram State machine diagram	Sec 2.3 (UC Chap 5)	
Week 4	Feb. 11, 13	OO frameworks 2D graphics in Java	Sec 3.3, 4.7, 5.5, 8.1 (CJv1 Sec 10.3)**	HW2
Week 5	Feb. 18, 20	Interfaces and lambdas	Sec 4.4.7 (CJv1 Chap 6)	
Week 6	Feb. 25, 27	Other Java features Documenting code (Javadoc)	Sec 4.5-4.6 (CJv1 Sec 4.7, Chap 7) Sec 6.1.2, App B (CJv1 Sec 4.9)	
Week 7	Mar. 4, 6	Inheritance Class design	Sec 5.1-5.4 (CJv1 Sec 5.1) Sec 6.1, 6.3 (CJv1 Sec 5.2)	
Week 8	Mar. 11, 13	<b>Exam 1</b> Unit testing (JUnit)	Handout or online reading	
Week 9	Mar. 18, 20	Spring break (no classes)		
Week 10	Mar. 25, 27	GUI framework	Sec 8.3 (CJv1 Chap 10-11)	HW3
Week 11	Apr. 1, 3	GUI framework Design by abstraction	Chap 7	
Week 12	Apr. 8, 10	Design by abstraction		
Week 13	Apr. 15, 17	Network programming	Sec 12.1 (CJv2 Chap 4)	HW4
Week 14	Apr. 22, 24	Concurrent programming	Chap 11 (CJv1 Chap 12)	
Week 15	Apr. 29, May 1	Collections and streams I/O framework	Sec 8.2 (CJv1 Chap 9) Sec 8.4 (CJv2 Chap 2/CJ9 Chap 9)	
Week 16	May 6, 8	Other topics or coding work		
Week 17	May 15	<b>Final</b> at 10:00 am-12:45 pm		

\*[UC] Martina Seidl, et al., *UML@Classroom: An Introduction to Object-Oriented Modeling*, Springer, 2015.

\*\*[CJv1] Cay S. Horstmann, *Core Java Volume I – Fundamentals*, 11<sup>th</sup> edition, Prentice Hall, 2018.

[CJv2] Cay S. Horstmann, *Core Java Volume II – Advanced Features*, 10<sup>th</sup> edition, Prentice Hall, 2016.

[CJ9] Cay S. Horstmann, *Core Java SE 9 for the Impatient*, 2<sup>nd</sup> edition, Addison-Wesley, 2017.

## Important Dates

January 21:	Dr. Martin Luther King, Jr. holiday (university closed)
January 22:	Classes begin
February 6:	Census day
March 11:	Exam 1
March 18-22:	Spring break (no classes)
March 29:	Cesar Chavez holiday (no classes)
April 5:	Course drop/withdrawal deadline
May 9:	Last day of classes
May 10:	Dead day
May 15:	Final on Wednesday at 10:00 am – 12:45 pm