CS 4390 Special Topics
CS 5381 Topics in Software Engineering:
Introduction to Formal Methods
Fall 2016

CRN: 18404 (CS 5381), 18405 (CS 4390)
Lecture: MW 1:30-2:50 pm in LART 319
Website: http://www.cs.utep.edu/cheon/cs5381 or cs4390
Instructor: Yoonsik Cheon (x-8028, ycheon@utep.edu); office hours: MW 3:00-4:00 pm in CCSB 3.0606
Prerequisite: CS 2302 for CS 4390

Description

The fundamental goal of this course is to raise the competence of software developers to create reliable software by improving the ability of students in precisely modeling, specifying, and reasoning about the correctness of computer programs. Considering the ubiquitousness of software and the frequency of software failures, the area of software correctness is an important part of the education of computer scientists and software engineers. However, there is a real concern with the lack of rigor and accountability in computer programming and software engineering, and the research agenda for software engineering states the need for strengthened mathematical foundation in the work force. The problem is not new, as shown by the following observation made in 1990[1].

> [there is] a fundamental difference between software engineers and other engineers. Engineers are well trained in the mathematics necessary for good engineering. Software engineers are not trained in the disciplines necessary to assure high quality software. ... The problem is not so much not having the mathematics necessary to solve the software problem, but instead having the trained software engineers.

Parnas once said that "*Professional engineers are expected to use discipline, science, and mathematics to assure that their products are reliable and robust. We should expect no less of anyone who produces programs professionally.*" This course will provide a small step toward realizing Parnas's by introducing formal methods---mathematically based techniques---and formal program verification---proving, mathematically, that a program agrees with its specification. It is essential for developers to employ these methods that offer high degree of assurance that the system's requirements are accurately capture the user's critical requirements and that an implementation in software is an accurate realization of the design. The specific topics to be covered in this course include:

- Concepts of formal methods
- Representative formal specification languages: OCL, Z, Larch, JML, and CSP
- Application of formal methods

Learning Objectives

- Understand the concepts of formal methods.
- Know representative formal specification languages.
- Precisely model and specify small software systems using formal specification languages.
- Compare and evaluate different formal methods to choose a suitable one for a given application.
- Apply formal verification techniques to code with low complexity to reason about its correctness.

.

[1] Cherniavsky, J. C. Software failures attract congressional attention. *Computing Research News*, 2(1):4–5, January 1990.

Textbook

There is no required textbook. Tutorials, introductory articles and research papers, reference manuals, and various on-line documents will be used as course material (see the suggested readings at the end of this syllabus). Hard copies of required readings will be provided.

Examinations

For CS 4390 students, there will be two exams: mid-term and final. The mid-term exam will take place during the regular class session and will be 80 minutes in length. The final will be a take-home exam in the last week (see page 4 for tentative exam dates).

For CS 5381 student, there will be only a mid-term exam; there will be no final exam. The mid-term exam will take place during the regular class session and will be 80 minutes in length.

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 all circumstances, you are required to provide official documentation before a make-up will be administered.

Assignments

There will be occasional written homework assignments (see page 4 for planned homework). All assignments shall be done individually unless otherwise specified, and no late submission will be accepted unless arrangements have been made in advance or unless unusual circumstances warrant an exception.

For CS 5381 students, there is another type of assignments: paper presentation. Students are expected to read and present research papers related to the course topics---formal methods and program verification. The number of presentations will be one or two depending on the class size. A suggested, tentative list of papers is found at the end of this syllabus, however students are free to choose papers for their presentations. The paper presentation is optional for CS 4390 students but will earn bonus points.

Projects

CS 5381 students should do a small, semester-long class project. The purpose of this project is to apply the concepts, techniques, methods learned from the course to your thesis or dissertation research. Sample project topics will be suggested by the instructor, but you may choose your own project topic; however, your topic must be approved by the instructor. You are expected to write a project proposal, submit a final project report, and present the project result in class.

CS 4390 students are not required, though encouraged, to do a semester project, but they can opt for the semester project instead of taking the final exam. If you do a semester project, you will earn more points toward your final grade.

Grading

Your grade is independent of anyone else's grade; that is, we do not grade on a curve. Everyone can get an A in this course. The purpose of grading is not to rank you, but to uphold a standard of quality and to give you feedback. The final letter grade will be based on a combination of assignments, project, exams, and class participation. The approximate percentages are as follows:

Assignment:	50%
Mid-term exam:	20%
Final exam/project:	30%

There are also up to 5% bonus points for lecture attendance and class participation. To earn this, 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:	A
80-89%:	B
70-79%:	C
60-69%:	D
below 60%:	F

The instructor reserves 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 mandatory. Your success in the course will improve greatly by attending classes. If you miss classes, it is your responsibility to keep up to date with lecture notes, assignments, and projects. The following is excerpted from the 2016-2017 Graduate/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, 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.

Tentative Schedule

The following table shows a tentative schedule of the course; refer to the course website for an up-to-date schedule.

Dates		Topics	Readings	Assignments
Week 1	Aug. 22, 24	Intro. and use case diagram Class diagram		
Week 2	Aug. 29, 31	Intro. to formal methods	[Wing90]	
Week 3	Sep. 5, 7	*No class – Labor day* Intro. to formal methods		
Week 4	Sep. 12, 14	Z	[Spivey89]	Homework 1
Week 5	Sep. 19, 21	Z Object Constraint Language (OCL)	[Warmer-Kleppe99]	
Week 6	Sep. 26, 28	OCL		Homework 2
Week 7	Oct. 3, 5	*Paper presentation: Z, OCL*+ Tabular notation	[Janicki-Parnas-Zucker96]	
Week 8	Oct. 10, 12	Tabular notation Java Modeling Language (JML)	[Leavens-Baker-Ruby06]	
Week 9	Oct. 17, 19	Project proposal+ **Exam 1**		Proposal+
Week 10	Oct. 24, 26	JML		Homework 3
Week 11	Oct. 31, Nov. 2	JML *Paper presentation: Tabular, JML*+		
Week 12	Nov. 7, 9	Algebraic: Larch	[Guttag-Horning86]	Homework 4
Week 13	Nov. 14, 16	Larch Process: CSP	[Hoare78]	
Week 14	Nov. 21, 23	CSP *Paper presentation: Larch, CSP*+		
Week 15	Nov. 28, 30	Final: take-home* Project work and presentation+		Final report+
Week 16	Dec. 7	**Final exam week**		

*Only for CS 4390 students

+Required for CS 5381 students and optional for CS 4390 students

Important Dates

August 22:	Class begins
September 5:	Labor day – University closed
September 7:	Census day
October 19:	Exam 1
October 28:	Course drop/withdrawal deadline
November 24-25:	Thanksgiving holiday - University closed
December 2:	Dead day
December 7:	Final on Wednesday at 4:00 pm – 6:45 pm

Required Readings

The following is a tentative list of required readings, and you are welcome to suggest additional readings.

[Wing90] Jeannette M. Wing. A Specifier's Introduction to Formal Methods. *IEEE Computer*, 23(9):8-24, September, 1990.

[Spivey89] J.M. Spivey. An Introduction to Z and Formal Specifications. *Software Engineering Journal*, 4(1):40-50, January, 1989.

[Warmer-Kleppe99] Jos Warmer and Anneke Kleppe. OCL: The Constraint Language of the UML, *Journal of Object-Oriented Programming*, 2(2):10-13, May 1999.

[Janicki-Parnas-Zucker96] Ryszard Janicki, David L. Parnas, and Jeffery Zucker. Tabular Representations in Relational Documents. In *Relational Methods in Computer Science*, Springer-Verlag, 1996.

[Leavens-Baker-Ruby06] Gary T. Leavens, Albert L. Baker, and Clyde Ruby. Preliminary Design of JML: A Behavioral Interface Specification Language for Java. *ACM SIGSOFT Software Engineering Notes*, 31(3):1-38, May 2006.

[Guttag-Horning86] J. Guttag and J. Horning, Report on the Larch Shared Language, *Science of Computer Programming*, 6:103-134, 1986.

[Hoare78] C.A.R. Hoare. Communicating Sequential Processes. *Communications of the ACM*, 21(8):666-677, August 1978.

Supplementary Readings

[Brinksma86] Ed Brinksma. A Tutorial on LOTOS. In *Protocol Specification, Testing, and Verification*, V, pages 171-194, Elsevier, 1986.

[Cheon-Vela10] Yoonsik Cheon and Melisa Vela. *A Tutorial on Functional Program Verification*, Technical Report 10-26, Department of Computer Science, University of Texas at El Paso, El Paso, TX, September 2010.

[Hamie04] A. Hamie. Translating the Object Constraint Language into the Java Modeling Language, *ACM Symposium on Applied Computing, Nicosia, Cyprus, March 14 -17, 2004*, pages 1531–1535, 2004.

[Hennicker-Hussmann-Bidoit02] Rolf Hennicker, Hinrich Hussmann, and Michel Bidoit. On the Precise Meaning of OCL Constraints. In T. Clark and J. Warmer, editors, *Object Modeling with the OCL, volume 2263 of Lecture Notes in Computer Science,* pages 69-84, Springer, 2002.

[Parnas93] David L. Parnas. Predicate Logic for Software Engineering. *IEEE Transactions on Software Engineering*, 19(9):856-862, September 1993.

[Warmer-Kleppe03] Jos Warmer and Anneke Kleppe. *The Object Constraint Language*, second edition, Addison-Wesley, 2003.

[Woodcock89] J.C.P. Woodcock. Structuring Specifications in Z. *Software Engineering Journal*, 4(1):51-66, January, 1989.