CS 5381: Topics in Software Engineering
Introduction to Formal Methods and Program Verification
Fall 2013

CRN: 16669
Lecture: MW 3:00-4:20 pm in HUDS 100 (Hudspeth Hall)
Website: http://www.cs.utep.edu/cheon/cs5381
Instructor: Yoonsik Cheon (x-8028, ycheon@utep.edu); office hours: MW 1:30-2:30 pm in CCSB 3.0606
Prerequisite: Department approval

Description
The fundamental goal of this course is to raise the competence of software developers to create and verify 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 correctness verification 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:

> [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:

- Basic concepts of formal methods and program verification
- Representative formal specification languages, such as Z, OCL, Larch, JML, and CSP
- Cleanroom-style functional program verification
- Application of formal methods

Learning Objectives
- Understand the concepts, principles, and theories of formal methods.
- Understand the basic concepts of formal program verification.
- Know representative formal specification languages.
- Know proof rules of basic control structures such as assignments, sequences, branches, and loops.
- Using a formal specification language, precisely model or specify software systems of small size.
- Apply formal verification techniques to code with low complexity to reason about its correctness.
- Compare and evaluate different methods to choose a suitable one for a given application or problem.

---

Textbook
There is no required textbook for this course. Tutorials, (introductory) research papers, reference manuals, and various on-line documents will be used as course material; see the suggested readings at the end of this document.

Examinations
There will be one mid-term exam and a final exam. The mid-term exam will take place during the regular class session and will be 80 minutes in length.

Assignments
There will be two kinds of assignments: in-class presentation and written homework assignments. 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. The suggested, tentative list of papers is found at the end of this document. There will be occasional written homework assignments. All assignments shall be done individually unless otherwise specified, and no late submission will be accepted.

Projects
You 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.

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:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>30%</td>
</tr>
<tr>
<td>Project</td>
<td>40%</td>
</tr>
<tr>
<td>Exam</td>
<td>30%</td>
</tr>
</tbody>
</table>

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

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 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.
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 (assignments, 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 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.

<table>
<thead>
<tr>
<th>Dates</th>
<th>Topics</th>
<th>Readings</th>
<th>Assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>Aug. 26, 28</td>
<td>Intro. to formal methods</td>
<td>[Wing90]</td>
</tr>
<tr>
<td>Week 2</td>
<td>Sep. 2, 4</td>
<td>No class - Labor day</td>
<td>[Spivey89]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Z</td>
<td>[Woodcock89]</td>
</tr>
<tr>
<td>Week 3</td>
<td>Sep. 9, 11</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>Week 4</td>
<td>Sep. 16, 18</td>
<td>Object Constraint Language (OCL)</td>
<td>[Warmer-Kleppe99]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Warmer-Kleppe03]</td>
</tr>
<tr>
<td>Week 5</td>
<td>Sep. 23, 25</td>
<td>OCL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Paper presentations: Z, OCL</td>
<td></td>
</tr>
<tr>
<td>Week 6</td>
<td>Sep. 30, Oct. 2</td>
<td>Tabular notation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Janicki-Parnas-Zucker96]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Homework 2</td>
</tr>
<tr>
<td>Week 7</td>
<td>Oct. 7, 9</td>
<td>Project proposal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exam 1</td>
<td></td>
</tr>
<tr>
<td>Week 8</td>
<td>Oct. 14, 16</td>
<td>Java Modeling Language (JML)</td>
<td>[Leavens-Baker-Ruby06]</td>
</tr>
<tr>
<td>Week 9</td>
<td>Oct. 21, 23</td>
<td>JML</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Paper presentations: Tabular, JML</td>
<td></td>
</tr>
<tr>
<td>Week 10</td>
<td>Oct. 28, 30</td>
<td>Algebraic: Larch</td>
<td>[Guttag-Horning86]</td>
</tr>
<tr>
<td>Week 11</td>
<td>Nov. 4, 6</td>
<td>Algebraic Process: CSP, Lotos</td>
<td>[Hoare78]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[Brinksma86]</td>
</tr>
<tr>
<td>Week 12</td>
<td>Nov. 11, 13</td>
<td>Process</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Paper presentations: Algebraic, process</td>
<td></td>
</tr>
<tr>
<td>Week 13</td>
<td>Nov. 18, 20</td>
<td>Intro. to program verification</td>
<td>[Hoare69]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hoare logic</td>
<td>[Cheon-Vela10]</td>
</tr>
<tr>
<td>Week 14</td>
<td>Nov. 25, 27</td>
<td>Functional program verification</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Paper presentations: Prog. verification</td>
<td></td>
</tr>
<tr>
<td>Week 15</td>
<td>Dec. 2, 4</td>
<td>Project presentations</td>
<td></td>
</tr>
<tr>
<td>Week 16</td>
<td>Dec. 9</td>
<td><strong>Final exam</strong> at 1:00 – 3:45 pm</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Important Dates

August 26: Class begins
September 2: Labor day – university closed
September 11: Census day
October 9: Exam 1
November 1: Course drop deadline
November 28-29: Thanksgiving holiday - university closed
December 5: Last day of classes
December 6: Dead day
December 9: Final on Monday at 1:00 – 3:45 pm
Readings
The following is a tentative list of readings. The list is subject to change, and you will be asked to add your own reading list; an up-to-date list will be available from the course website.


