Automated Software Engineering (CSE6323)

University of Texas at Arlington
Computer Science and Engineering

Instructor: *Taylor Johnson*
Email Address: *firstname.lastname@uta.edu* (replace firstname with Taylor and lastname with Johnson and note there is no mavs subdomain)
Office Number: Engineering Research Building 559 (ERB 559)
Office Telephone Number: 817-272-3610 (note: voicemail is never checked, email me if you miss me)
Faculty Profile: *http://www.taylortjohnson.com/*
Section Information: CSE 6323-001, Fall 2015 (*TBD*)
Course Website: *http://www.taylortjohnson.com/class/cse6323/f15/*
Time and Place of Class Meetings: Tuesdays/Thursdays 3:30pm-4:50pm, UTA - Woolf Hall 210 (WH 210)
Office Hours: ERB559, Tuesdays/Thursdays 10am-11am and by appointment (email me to schedule by appointment)

Graduate Teaching Assistant (GTA): TBD
Graduate Teaching Assistant Email Address:
Graduate Teaching Assistant Office Hours: TBD, ERB 562

Course Schedule (*Syllabus*; note that all information appearing on this website supersedes that appearing in the syllabus PDF, that is, this website is more up-to-date): The instructor for this course reserves the right to adjust this schedule in any way that serves the educational needs of the students enrolled in this course. All assignments and due dates are approximate at this point and will be updated on this website as the semester progresses. All readings refer to the Alur required textbook unless otherwise noted.

Date	Content	Resources / Readings	Assignments
8/27	Introduction and Course Overview: What is automated software engineering and what are formal methods? (First Day of Classes)	Alur Chapter 1	
9/1	Math review: propositional logic and set theory	Reading: Aho and Ullman, Chapter 12 (particularly 12.2, 12.3, and 12.4, remainder may be skimmed, except skip 12.6, as they will come up again later in the course) and Chapter 14 (14.1, 14.2, 14.3, 14.4); Additional References: Lee and Seshia, Appendix B	Homework 1 Assigned (files on BlackBoard)
9/3	Math review: first-order logic and higher-order logic	Reading: Aho and Ullman, Chapter 12 (particularly 12.2, 12.3, and 12.4, remainder may be skimmed, except skip 12.6, as they will come up again later in the course) and Chapter 14 (14.1, 14.2, 14.3, 14.4); Additional References: Lee and Seshia, Appendix B	
9/8	Math review: automata theory and formal languages	Reading: Aho and Ullman, Chapter 12 (particularly 12.2, 12.3, and 12.4, remainder may be skimmed, except skip 12.6, as they will come up again later in the course) and Chapter 14 (14.1, 14.2, 14.3, 14.4); Additional References: Lee and Seshia, Appendix B	
			Homework 1

9/10	Formal models: transition systems	Reading: Alur, Chapters 2 and 4 (skim chapter 4); Additional References: Lee and Seshia, Chapters 3 and 5 (and partly 6)	Due; Homework 2 Assigned (files on BlackBoard)
9/15	Formal models: synchronous networks and transition systems	Reading: Alur, Chapters 2 and 4 (skim chapter 4); Additional References: Lee and Seshia, Chapters 3 and 5 (and partly 6)	
9/17	Formal specifications: safety	Reading: Alur, Section 3.1, Section 5.1; Additional References: Lee and Seshia, Chapter 13	
9/22	Formal methods for safety: inductive invariant proofs	Reading: Alur, Section 3.2	Homework 2 Due; Homework 3 Assigned (files on BlackBoard)
9/24	Formal methods for safety: reachability analysis	Reading: Alur, Section 3.3	
9/29	Formal models: asynchronous networks and transition systems	Reading: Alur, Section 5.2	
10/1	Formal specifications: liveness, linear temporal logic (LTL), and computation tree logic (CTL)	Reading: Alur, Section 5.2	
10/6	Formal methods for liveness: LTL model checking and repeatability analysis	Reading: Alur, Section 5.3	Homework 4 Assigned (files on BlackBoard)
10/8	Automated/interactive theorem proving: overview	Klein and Nipkow, Chapter 13	Homework 3 Due
10/13	Automated/interactive theorem proving: inductive invariant proofs	Klein and Nipkow, Chapter 13	Homework 4 Due; Programming Assignment 1 Assigned
10/15	Automated/interactive theorem proving: liveness proofs with ranking functions	Alur, Section 5.4; Klein and Nipkow, Chapter 13	
10/20	Timed automata	Alur, Chapter 7.1-7.2	
10/22	Model checking timed automata	Alur, Chapter 7.3	
10/27	Hybrid automata	Alur, Chapter 9	
10/29	Model checking hybrid automata	Alur, Chapter 9	
11/3	Satisfiability and satisfiability modulo theories (SMT)	Z3	Programming Assignment 1 Due; Programming Assignment 2 Assigned

11/5	Bounded model checking	Z3	
11/10	Bounded model checking for timed and hybrid automata	Z3	
11/12	Test case generation with SAT/SMT	Z3	
11/17	Binary Decision Diagrams (BDDs); Guest Lecture/No class (Instructor traveling to NSF)	nuXmv	
11/19	Data Structures of SAT/SMT; Guest Lecture/No Class (Instructor Traveling to AFRL)	Z3	Programming Assignment 2 Due; Final Project Assigned
11/24	No class (Thanksgiving Holiday)		
11/26	No Class (Thanksgiving Holiday)		
12/1	Difference Bound Matrices (DBMs); Guest Lecture/No Class (Instructor Traveling to RTSS)	Uppaal	
12/3	potpourri: k-Induction, Assume-Guarantee Reasoning; Guest Lecture/No Class (Instructor Traveling to RTSS)	nuXmv	
12/8	Course review (Last Day of Class)		
12/9	Semester Last Day of Classes		Final Project Due
12/17, 2pm-4:30pm	Final Exam (*Scheduled Final Exam Time*)		

Required Textbook and Other Course Materials:

- Main Required Textbook: "*Principles of Cyber-Physical Systems*," Rajeev Alur, MIT Press, First Edition, 2015; *a digital copy may be purchased here* or a *print copy can be found on Amazon*

Optional Textbooks and Additional References (free online):

- Background math: "*Foundations of Computer Science*," Al Aho and Jeff Ullman, 1994.
- Alternative main textbook: "*Introduction to Embedded Systems: A Cyber-Physical Systems Approach* ," Second Edition, Edward Lee and Sanjit Seshia
- Theorem proving: "*Concrete Semantics with Isabelle/HOL*," Tobias Nipkow and Gerwin Klein, 2015
- *nuXmv user manual*
- *NuSMV user manual*

Software Tools: We will use a variety of automated software engineering tools in this course, include test case generation, model checkers, SAT/SMT solvers, and automated/interactive theorem provers integrated with programming languages, such as through the Frama-C static analysis tool's links to Coq, PVS, and Isabelle. In particular, we plan to use the following tools and possibly others, especially dependent upon student interest in various application domains and target languages (e.g., C, Java, .NET, Python, etc.):

- Model checkers: *nuXmv (newest version)* / *NuSMV*, *Spin*, *Uppaal*
- SAT/SMT solvers: *Z3*
- Static analysis tools: *Frama-C*
- Dynamic analysis tools: *Daikon*
- Automated/interactive theorem provers: *Isabelle*, and possibly *PVS* or *Coq*
- Test case generation tools: *Pex*, *Simulink Design Verifier*

Optional Textbooks and Additional References (for purchase):

- "*Model Checking*," Edmund Clarke, Orna Grumberg, and Doron Peled, MIT Press, 1999.
- "*Principles of Model Checking*," Christel Baier and Joost-Pieter Katoen, MIT Press, 2008

Description of Course Content: Study of foundations, techniques and tools for automating software processes and methodologies including analysis, design, implementation, testing, and maintenance of large software systems.

Prerequisites: Prerequisite: CSE 5324 or consent of instructor. Students are expected to be proficient with elementary computer science theory (discrete math, algorithms, graphs, etc.), discrete mathematics, and software development. Students are expected to have working experiences with software development, including software version control systems such as Git, Mercurial, and/or Subversion.

Student Learning Outcomes: The objective of this course is to introduce graduate computer science and engineering students to methods for automated analysis of software systems, in particular, using automated formal methods, such as model checking, model checking-inspired test case generation, satisfiability, theorem proving, etc. At course conclusion, students should be able to:

- Understand propositional, first-order, and higher-order logic and their relationships to set theory.
- Understand formal specifications such as linear temporal logic (LTL), invariants, computation tree logic (CTL), etc.
- Be able to write formal specifications from informal (plain English) requirements for software systems
- Understand formal models such as extended state machines, hybrid automata, timed automata, and transition systems
- Understand syntax and semantics for models of software systems
- Use automated software engineering tools, such as test case generation tools, model checkers, SAT/SMT solvers, interactive theorem provers, etc.
- Understand the theoretical basis of algorithms and data structures used in automated software engineering tools such as model checkers, SAT/SMT solvers, etc.
- Understand static analysis and dynamic analysis
- Be profcient in deriving formal specifications and formal models for software systems

Descriptions of major assignments and examinations: Coursework may include homework assignments, several programming assignments/small projects using various automated software engineering tools like model checkers, projects, exams, and quizzes. Online and/or in-class quizzes and discussions will make up a portion of the grade. All assignments are subject to change and are not finalized, but will be determined based on student interest and progress. Approximate due dates of assignments are shown in the course schedule and are all subject to change.

Attendance: At the University of Texas at Arlington, taking attendance is not required. Rather, each faculty member is free to develop his or her own methods of evaluating students' academic performance, which includes establishing course-specific policies on attendance. As the instructor of this section, students are strongly encouraged to attend lectures, complete reading assignments, come to office hours, and make use of all available educational resources to ensure their educational progress.

Other Requirements: Exams will be closed book, but students will be allowed to bring a two-sided sheet of letter-size paper. Students are expected to check the course website for updates to the course schedule throughout the semester.

Grading: Grade percentages will be calculated based on the following weights:

- Homework and Programming Assignments: 40%
- Course Project: 20%
- Quizzes / Participation: 20%
- Final Exam: 20%

Letter grades will be determined based on the following ranges:

- 100 >= A >= 90
- 90 > B >= 80
- 80 > C >= 70
- 70 > D >= 60
- 60 > F >= 0

The instructor reserves the right to move the thresholds down based on the distribution of final percentages, but they will not move up (e.g., if a grade percentage is between 90 and 100, this will receive an A). Students are expected to keep track of their performance throughout the semester and seek guidance from available sources (including the instructor) if their performance drops below satisfactory levels.

Make-Up Assignments, Exams, and Late Assignment Submission: If you miss an exam or quiz due to unavoidable circumstances (e.g., health), you must notify the instructor in writing via email as soon as possible and request a makeup approval. If it is a planned (non-emergency) abcense, you must inform the instructor ahead of time! Do NOT ask for make-ups if you do not complete something due to travel (except when you are required to travel to represent the university or department on official business, but request at least 3 days ahead of the due date or exam time). If you submit an assignment late, it will have points taken off at a rate of 33% per day.

Grade Grievances: Any appeal of a grade in this course must follow the procedures and deadlines for grade-related grievances as published in the current undergraduate catalog (see *here*).

The first step is as follows. If you do not believe a grade on a particular assignment is correct, you may appeal the grade in writing (by email) within 5 days. Grade appeals must be appealed to the appropriate GTA first, then to the instructor if necessary.

Drop Policy: Students may drop or swap (adding and dropping a class concurrently) classes through self-service in MyMav from the beginning of the registration period through the late registration period. After the late registration period, students must see their academic advisor to drop a class or withdraw. Undeclared students must see an advisor in the University Advising Center. Drops can continue through a point two-thirds of the way through the term or session. It is the student's responsibility to officially withdraw if they do not plan to attend after registering. Students will not be automatically dropped for non-attendance. Repayment of certain types of financial aid administered through the University may be required as the result of dropping

classes or withdrawing. For more information, contact the *Office of Financial Aid and Scholarships*.

Americans with Disabilities Act: The University of Texas at Arlington is on record as being committed to both the spirit and letter of all federal equal opportunity legislation, including the Americans with Disabilities Act (ADA). All instructors at UT Arlington are required by law to provide "reasonable accommodations" to students with disabilities, so as not to discriminate on the basis of that disability. Any student requiring an accommodation for this course must provide the instructor with official documentation in the form of a letter certified by the staff in the Office for Students with Disabilities, University Hall 102. Only those students who have officially documented a need for an accommodation will have their request honored. Information regarding diagnostic criteria and policies for obtaining disability-based academic accommodations can be found *here* or by calling the Office for Students with Disabilities at 817-272-3364.

Title IX: The University of Texas at Arlington is committed to upholding U.S. Federal Law "Title IX" such that no member of the UT Arlington community shall, on the basis of sex, be excluded from participation in, be denied the benefits of, or be subjected to discrimination under any education program or activity. For more information, visit *www.uta.edu/titleIX*.

Academic Integrity: Students enrolled in this course are expected to adhere to the UT Arlington Honor Code: "I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence. I promise that I will submit only work that I personally create or contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code." UT Arlington faculty members may employ the Honor Code as they see fit in their courses, including (but not limited to) having students acknowledge the honor code as part of an examination or requiring students to incorporate the honor code into any work submitted. Per UT System Regents' Rule 50101, Section 2.2, suspected violations of university's standards for academic integrity (including the Honor Code) will be referred to the Office of Student Conduct. Violators will be disciplined in accordance with University policy, which may result in the student's suspension or expulsion from the University.

Academic Integrity Violations, Cheating, Plagiarism, Disallowed Collaboration, and Consequences: The instructor makes use of standard cheating detection tools including and not limited to text comparison of homework, source code, etc., the *Moss software plagiarism detection tool*, and others. If the instructor suspects cheating, plagiarism, disallowed collaboration, etc., *the instructor will submit the violations and evidence without exception to the university according to university policy*. There is zero tolerance for cheating and academic dishonesty. For your information, you should be aware of the types of suspected cheating and their consequences, which include: *failing the course, failing the assignment, and possible further consequences including expulsion, loss of scholarships / funding, etc.*.

Electronic Communication: UT Arlington has adopted MavMail as its official means to communicate with students about important deadlines and events, as well as to transact university-related business regarding financial aid, tuition, grades, graduation, etc. All students are assigned a MavMail account and are responsible for checking the inbox regularly. There is no additional charge to students for using this account, which remains active even after graduation. Information about activating and using MavMail is available *here*.

Student Feedback Survey: At the end of each term, students enrolled in classes categorized as "lecture," "seminar," or "laboratory" shall be directed to complete an online Student Feedback Survey (SFS). Instructions on how to access the SFS for this course will be sent directly to each student through MavMail approximately 10 days before the end of the term. Each student's feedback enters the SFS database anonymously and is aggregated with that of other students enrolled in the course. UT Arlington's effort to solicit, gather, tabulate, and publish student feedback is required by state law; students are strongly urged to participate. For more information, visit *here*.

Final Review Week: A period of five class days prior to the first day of final examinations in the long sessions shall be designated as Final Review Week. The purpose of this week is to allow students sufficient time to prepare for final examinations. During this week, there shall be no scheduled activities such as required field trips or performances; and no instructor shall assign any themes, research problems or exercises of similar scope that have a completion date during or following this week unless specified in the class syllabus. During Final Review Week, an instructor shall not give any examinations constituting 10% or more of the final grade, except makeup tests and laboratory examinations. In addition, no instructor shall give any portion of the final examination during Final Review Week. During this week, classes are held as scheduled. In addition, instructors are not required to limit content to topics that have been previously covered; they may introduce new concepts as appropriate.

Emergency Exit Procedures: Should we experience an emergency event that requires us to vacate the building, students should exit the room and move toward the nearest exit, which is located to either the west or east sides of Woolf Hall to the southwest or east stairwells. Emergency exit maps are available for all buildings at *https://www.uta.edu/campus-ops/ehs/fire/Evac_Maps_Buildings.php* and for this classroom at *https://www.uta.edu/campus-ops/ehs/fire/Evac_Maps_All/Evac_WH/Evac_WH_210.pdf*. When exiting the building during an emergency, one should never take an elevator but should use the stairwells. Faculty members and instructional staff will assist students in selecting the safest route for evacuation and will make arrangements to assist handicapped individuals.

Student Support Services: UT Arlington provides a variety of resources and programs designed to help students develop academic skills, deal with personal situations, and better understand concepts and information related to their courses. Resources include tutoring, major-based learning centers, developmental education, advising and mentoring, personal counseling, and federally funded programs. For individualized referrals, students may visit the reception desk at University College (Ransom Hall), call the Maverick Resource Hotline at 817-272-6107, send a message to *resources@uta.edu*, or view the information at *www.uta.edu/resources*.

Emergency Phone Numbers: In case of an on-campus emergency, call the UT Arlington Police Department at 817-272-3003 (non-campus phone), 2-3003 (campus phone). You may also dial 911.

Last modified: August 24, 2015 22:17:43.