NAU NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing, and Cyber Systems

CS 499 **Testing Embedded Systems** Syllabus

Term	Class No.	Section	Units	Days & Times	Room	Mode
Spring 2020	CS 599	001	3	Not known (once per week evening)	N/A	Face-to-face

Enrollment Requirements

Pre-requisite: CS 200 or equivalent, substantial experience in C/C++ programming, graduate standing

Course Website

https://github.com/agroce/cs499_embedded_testing

Instructor(s)

Dr. Alex Groce
Email: Alex.Groce@nau.edu
Slack: TBA
Office Hours: TBA

Course Description:

Intensive study of the underlying concepts, algorithms, and software tooling associated with testing of embedded and low-level software systems, in for particular safety-critical or security-critical contexts. Letter grade only.

Course Purpose

This course covers the basic and advanced topics in (automated) testing for embedded systems and software that interfaces with hardware systems. There is a large overlap with the general topic of "testing systems software." Topics covered include challenges of testing low-level C/C++ code, advantages and disadvantages of simulating hardware, automated test generation for C and C++, model checking and formal verification of critical routines in low-level code, bug types most associated with low-level and embedded code, and case studies of testing/analysis of embedded systems code. We explore cutting-edge research topics in this domain, in particular test generation and specification problems related to low-level languages without memory safety guarantees or testing-friendly error mechanisms such as exceptions. This course prepares students for roles in embedded systems research and development, and for working with complex low-level systems software in general. This class is an elective in the CS program and can help fulfill required elective units.

Course Student Learning Outcomes

Upon successful completion of this course, students will be able to demonstrate the following competencies:

- **LO1**: Understand and explain the unique challenges of testing low-level, systems, and embedded software
- **LO2**: Select, synthesize, and evaluate tools that support testing low-level, systems, and embedded software, especially coverage-driven mutational fuzzers, as well as more complete but less scalable tools, and understand the research community differences between different types of tooling.
- **LO3**: Select and synthesize algorithms and data structures for generating inputs exhaustively when feasible, or via partition/equivalence classes;
- **LO4**: Select and synthesize programming techniques that support testability of code and fuzzing of interface boundaries; and
- **LO5**: Synthesize, apply, and analyze results of data analyses and draw inferences on comparative effectiveness of various testing approaches, using coverage and mutation-based evaluations.
- **LO6**: Understand and apply results from cutting-edge academic and industrial research on fuzzing, test input generation, binary analysis, and other advanced topics in low-level test generation.
- **LO7**: Select and synthesize connections between other research fields, such as programming language theory, control theory, and machine learning, and test generation and execution problems in low level languages/settings.

Assignments / Assessments of Course Student Learning Outcomes

Learning outcomes are assessed through a variety of means: A midterm and second exam assess student ability to describe and explain foundational concepts in testing of low-level and embedded systems, including mutation-based coverage-driven fuzzing and bounded model checking of C code. Graduate student responses are expected to reach a higher level of depth and breadth than the responses of their undergraduate counterparts, which will be reflected in additional questions on both tests.

Individual programming assignments assess student ability to synthesize and analyze low-level and embedded systems testing concepts and methods (LO2, LO3, LO4, and LO5). Key programming assignments will include: fuzzing a simple function with libFuzzer and AFL; using DeepState to fuzz a multi-function API; implementing a simple hardware simulator; using CBMC to verify the behavior of a complex bit-manipulation function. A final, larger programming assignment will require using the explicit-state model checker SPIN to execute and generate tests for a C program.

A final project will address Quantitative Reasoning and Scientific Inquiry by requiring students to use multiple input generation methods to target a real-world piece of code, and report on the measured effectiveness (and speculate as to the causes of differences) between methods. At least one applied method must be original, modifying the behavior of an existing tool or algorithm in some way that is tied to the research literature in a principled way, but has not previously been investigated. The connection to research literature must be thoroughly documented and justified, and the statistical power and other threats to validity of the experiments must be properly explained.

Grading System

A weighted sum of assessment components is used to determine your final grade in the course:

- Programming Assignments: **30%**
- Final Project: **30%**
- Midterm Exam: **20%**
- Second Exam: **20%**

Grades will be assigned using the weighted sum described above using this scale:
A ≥ 90%, **B** ≥ 80%, **C** ≥ 70%, **D** ≥ 60%, **F** < 60%.

There is no “curve”. Each student’s grade is based on their own outcomes assessments and not affected by the grades of other students. Extra credit opportunities may present themselves throughout the semester and will be announced during class meetings. Mistakes in grading to happen, and students are encouraged to discuss such concerns with the instructor during office hours.

Readings and Materials

Students will not be required to purchase a textbook for this class.

The “textbook” is the online, interactive text “Generating Software Tests” by Zeller, Gopinath, Bohme, Fraser, and Holler (https://www.fuzzingbook.org/). There will also be readings from *Introduction to Embedded Systems: a Cyber-Physical Systems Approach* by Lee and Seshia, which is freely available as a PDF download from the MIT press and from UC Berkeley. Additional readings will be papers and blog posts also available online, including "The Mars Rover Spirit FLASH Anomaly" by Reeves and Neilson, "Randomized Differential Testing as a Prelude to Formal Verification" by Groce, Holzmann, and Joshi, and various discussions of embedded systems testing by John Regehr and others, including Twitter threads.

Students will also need access to a Linux distribution (preferably a recent Ubuntu) to compile and run code, available as part of NAU's standard software distribution and most computer labs (particularly labs in 69-106 and 69-105). Students with Mac or Windows computers can use Docker images to support all required tasks.

We will use a dedicated Slack channel for the class to communicate in more informal ways that are similar to how developers in a real embedded development scenario would communicate.

Class Outline and Tentative Schedule

The course topics and a tentative schedule serve as an outline for the class:

Week 1	Introduction to the class, introduction to embedded systems testing
Week 2	Using AFL and libFuzzer to test a C function; challenges of building for fuzzing/instrumentation **(Programming Assignment 1)**
Week 3	Memory safety, sanitizers, and undefined behavior in C/C++
Week 4	A taxonomy and set of case studies on low-level/embedded bugs in real systems, with a focus on examples from NASA/space exploration
Week 5	Beyond single functions: fuzzing a complete API using DeepState; using multiple back-ends to generate tests **(Programming Assignment 2)**
Week 6	Hardware simulation, fault injection, hardware reliability expectations
Week 7	Case study: putting it all together to test a file system API; test-case reduction and bug triage; NASA file system example in detail; testfs from U Toronto
Week 8	**Midterm exam** and discussion of answers to midterm exam
Week 9	Symbolic execution and binary analysis of code, tradeoffs between scalability and completeness **(Programming Assignment 3)**
Week 10	Using coverage and mutation testing to evaluate fuzzing/testing efforts; advantages of ensemble fuzzing methods
Week 11	Static analysis: testing without running the code
Week 12	CBMC and complete (bounded) verification of C code; the difference between proof and testing **(Programming Assignment 4, Programming Assignment 5)**
Week 13	Security & side-channels, performance/real-time guarantees and other more complex discrete properties; hybrid-systems and continuous properties overview **(Final Project Due)**
Week 14	Review for second exam
Week 15	In-class second exam, discussion of exam answers and summary
Week 16	No final exam

Due dates for quizzes and homework will be posted on the github site.

Course Policies

The following policies will apply to this course:

- Students who have not completed the prerequisite(s) for this course, or who are absent from the class during the first week may be administratively dropped from the course.
- The makeup and late work policies are as follows:
    - Programming assignments: No make-ups or late submissions allowed.
    - Exams: Make-up exams will be given with approval from the instructor. Make-up exams may be considerably different than the original exam. Make-up exams must be taken within 3 business days of the original exam.
- Cheating and plagiarism are strictly prohibited. All academic integrity violations are treated seriously. All work you submit for grading must be your own. You are encouraged to discuss the intellectual aspects of assignments with other class participants. However, each student is responsible for formulating solutions on their own and in their own words. Academic integrity violations will result in penalties including, but not limited to, a zero on the assignment, a failing grade in the class, or expulsion from NAU. *See the NAU academic integrity policy:* https://www5.nau.edu/policies/Client/Details/307.
- Electronic device usage must support learning in the class. All cell phones, PDAs, music players and other entertainment devices must be turned off (or in silent mode) during lecture, and may not be used at any time. Laptops or workstations (if present) are allowed for note-taking and activities only during lectures; no web surfing or other use is allowed. I devote 100% of my attention to providing a high-quality lecture; please respect this by devoting 100% of your attention to listening and participating.
- Your final grade will be calculated in Excel using the grading system described above and then entered in LOUIE. Please check LOUIE for your final grade.
- Email to the instructor and teaching assistants must be respectful and professional. Specifically, all emails should:
    - Contain a salutation, (for example, "Dear Dr. Groce")
    - Contain a closing, (for example, "Best, Jane Doe")
    - The body should contain complete sentences and correct grammar including correct usage of lowercase and uppercase letters. Composing emails on a mobile device is **not** an excuse for poor writing.
    - The body of your message should also be respectful and explain the full context of the query.

- The subject should be prefixed with “INF110” so that the message can be easily identified or placed in an auto-folder. The subject should also use lower case and upper case correctly.
- Although email will typically be answered quickly, you should allow up to three (3) business days for a response.
- If you have a question that would require a long response or you have a lot of questions, please come to office hours or schedule an appointment with the instructor.

- Visiting the instructor(s) during office hours is encouraged! I am happy to talk about the class, careers, research, and topics related (even loosely) to this course.

Appendix A. UNIVERSITY POLICY STATEMENTS

ACADEMIC INTEGRITY

NAU expects every student to firmly adhere to a strong ethical code of academic integrity in all their scholarly pursuits. The primary attributes of academic integrity are honesty, trustworthiness, fairness, and responsibility. As a student, you are expected to submit original work while giving proper credit to other people's ideas or contributions. Acting with academic integrity means completing your assignments independently while truthfully acknowledging all sources of information, or collaboration with others when appropriate. When you submit your work, you are implicitly declaring that the work is your own. Academic integrity is expected not only during formal coursework, but in all your relationships or interactions that are connected to the educational enterprise. All forms of academic deceit such as plagiarism, cheating, collusion, falsification or fabrication of results or records, permitting your work to be submitted by another, or inappropriately recycling your own work from one class to another, constitute academic misconduct that may result in serious disciplinary consequences. All students and faculty members are responsible for reporting suspected instances of academic misconduct. All students are encouraged to complete NAU's online academic integrity workshop available in the E-Learning Center and should review the full academic integrity policy available at https://policy.nau.edu/policy/policy.aspx?num=100601.

COURSE TIME COMMITMENT

Pursuant to Arizona Board of Regents guidance (Academic Credit Policy 2-224), for every unit of credit, a student should expect, on average, to do a minimum of three hours of work per week, including but not limited to class time, preparation, homework, and studying.

DISRUPTIVE BEHAVIOR

Membership in NAU's academic community entails a special obligation to maintain class environments that are conductive to learning, whether instruction is taking place in the classroom, a laboratory or clinical setting, during course-related fieldwork, or online. Students have the obligation to engage in the educational process in a manner that does not breach the peace, interfere with normal class activities, or violate the rights of others. Instructors have the authority and responsibility to address disruptive behavior that interferes with student learning, which can include the involuntary withdrawal of a student from a course with a grade of "W". For additional information, see NAU's disruptive behavior policy at https://nau.edu/university-policy-library/disruptive-behavior.

NONDISCRIMINATION AND ANTI-HARASSMENT
NAU prohibits discrimination and harassment based on sex, gender, gender identity, race, color, age, national origin, religion, sexual orientation, disability, or veteran status. Due to potentially unethical consequences, certain consensual amorous or sexual relationships between faculty and students are also prohibited. The Equity and Access Office (EAO) responds to complaints regarding discrimination and harassment that fall under NAU's Safe Working and Learning Environment (SWALE) policy. EAO also assists with religious accommodations. For additional information about SWALE or to file a complaint, contact EAO located in Old Main (building 10), Room 113, PO Box 4083, Flagstaff, AZ 86011, or by phone at 928-523-3312 (TTY: 928-523-1006), fax at 928-523-9977, email at equityandaccess@nau.edu, or via the EAO website at https://nau.edu/equity-and-access.

TITLE IX
Title IX is the primary federal law that prohibits discrimination on the basis of sex or gender in educational programs or activities. Sex discrimination for this purpose includes sexual harassment, sexual assault or relationship violence, and stalking (including cyber-stalking). Title IX requires that universities appoint a "Title IX Coordinator" to monitor the institution's compliance with this important civil rights law. NAU's Title IX Coordinator is Pamela Heinonen, Director of the Equity and Access Office located in Old Main (building 10), Room 113, PO Box 4083, Flagstaff, AZ 86011. The Title IX Coordinator is available to meet with any student to discuss any Title IX issue or concern. You may contact the Title IX Coordinator by phone at 928-523-3312 (TTY: 928-523-1006), by fax at 928-523-9977, or by email at pamela.heinonen@nau.edu. In furtherance of its Title IX obligations, NAU will promptly investigate and equitably resolve all reports of sex or gender-based discrimination, harassment, or sexual misconduct and will eliminate any hostile environment as defined by law. Additional important information about Title IX and related student resources, including how to request immediate help or confidential support following an act of sexual violence, is available at http://nau.edu/equity-and-access/title-ix.

ACCESSIBILITY
Professional disability specialists are available at Disability Resources to facilitate a range of academic support services and accommodations for students with disabilities. If you have a documented disability, you can request assistance by contacting Disability

Resources at 928-523-8773 (voice), 928-523-6906 (TTY), 928-523-8747 (fax), or dr@nau.edu (e-mail). Once eligibility has been determined, students register with Disability Resources every semester to activate their approved accommodations. Although a student may request an accommodation at any time, it is best to initiate the application process at least four weeks before a student wishes to receive an accommodation. Students may begin the accommodation process by submitting a self-identification form online at https://nau.edu/disability-resources/student-eligibility-process or by contacting Disability Resources. The Director of Disability Resources, Jamie Axelrod, serves as NAU's Americans with Disabilities Act Coordinator and Section 504 Compliance Officer. He can be reached at jamie.axelrod@nau.edu.

RESPONSIBLE CONDUCT OF RESEARCH

Students who engage in research at NAU must receive appropriate Responsible Conduct of Research (RCR) training. This instruction is designed to help ensure proper awareness and application of well-established professional norms and ethical principles related to the performance of all scientific research activities. More information regarding RCR training is available at https://nau.edu/research/compliance/research-integrity.

SENSITIVE COURSE MATERIALS

University education aims to expand student understanding and awareness. Thus, it necessarily involves engagement with a wide range of information, ideas, and creative representations. In their college studies, students can expect to encounter and to critically appraise materials that may differ from and perhaps challenge familiar understandings, ideas, and beliefs. Students are encouraged to discuss these matters with faculty.

Updated 8/20/2018