

COM S 413/513 and CPR E 513: Program Analysis

Iowa State University

Fall 2023

Lectures: T/Th 11:00am-12:15pm, CARVER 0202

Instructor: Wei Le (weile@iastate.edu: please preface all email subjects with “413/513: ”)

Office Hours: T 12:15pm-1:15pm, Atanasoff 210

TA: Ashwin K J (ashwinkj@iastate.edu)

Office Hours:

Wed noon-12:30 pm, <https://iastate.webex.com/meet/ashwinkj>

Fri 3-4pm, Communications 1021

Lecture notes and homework assignments: <https://github.com/wei-le/programanalysiscourse>

Homework submissions and grades: Canvas

Q&A: Piazza

1 Course Description

Algorithms, AI techniques, and tools for automatically reasoning about code and program executions. Theory and foundations related to control flow analysis, dataflow analysis, abstract interpretation, and symbolic execution. Applications of program analysis to bug detection, test input generation, debugging, program repair, specification inference and trustworthy AI engineering. Concepts, algorithms, tools, benchmarks, methodologies for solving problems using program analysis and for preparing research in program analysis. —from Iowa State University course catalog.

2 Course Objectives

After accomplishing this course, students will

1. understand basic concepts and terminologies in program analysis,
2. know and implement program analysis algorithms,
3. learn the frontier of program analysis and software engineering research (especially for 513 students),
4. gain experience with open source environment and tools, and
5. improve problem solving skills to address software engineering challenges.

3 Prerequisites

COM S 327 or CPR E 288: C and UNIX coding experience is required

COM S 342: Any programming languages and compiler knowledge can be very helpful

4 Textbooks and Resources

The course does not have a required text book. We will use lecture notes and papers to teach and learn. The following is a list of additional materials for your references:

- *Compilers: Principles, Techniques, and Tools (second edition)*, by Alfred Aho, Jeffrey Ullman, Ravi Sethi and Monica Lam
- *Advanced Compiler Design and Implementation* by Steven Muchnick
- *Principles of Program Analysis* by Chris Hankin, Flemming Nielson, and Hanne Riis Nielson
- Jonathan Aldrich, CMU, 15-819 O Program Analysis
- Anders Møller and Michael I. Schwartzbach, Aarhus University, Static Program Analysis
- Monica Lam, Stanford, CS243 Program Analysis and Optimization
- Jens Palsberg, UCLA, CS232 Static Program Analysis
- Stephen Chong, Harvard, CS252r Advanced Topics in Programming Languages
- Mooly Sagiv, Tel Aviv University, Program Analysis

5 Tentative Schedule

Week	Topic	Homework Assignment
1	intro & bug detection	bug reproduction [interview]
2	control flow analysis	CFG and path [written]
3	dataflow analysis	an example of dataflow [written]
4	abstract intepretation	an example of abstract intepretation [written]
5	fuzzing	research paper [reading], final project team and idea
6	symoblic execution	run fuzzing & symbolic execution [project1]
7	midterm project and catchup	final project proposal (team of 3-4)
8	dependencies & slicing	an example of dependency and slicing [written]
9	delta debugging	implement delta-debugging [project2]
10	specification inference	Diakon invariants [interview]
11	software changes and versions	open-source program analysis tool [project3]
12-13	AI for program analysis tasks	research paper [reading]
14	analyzing and testing deep learning models	final project presentation & report
15	final project presentation	

6 Course Work and Evaluation

1. Program analysis paper reading (12%) → terminologies, learn frontier of program analysis and software engineering research
2. Homework assignments (58%):
 - 2 interviews (8%) → demo, conversation using program analysis terminologies and languages
 - 3 projects (30%) → implement program analysis algorithms, problem solving skills, gain experience of open source environment and tools
 - 4 written assignments (20%) → terminologies and concepts, know program analysis algorithms
3. Final Project (30%) → everything
 - 413: implement a program analysis tool
 - 513: develop a research prototype and writing a research report

Teamwork: interview questions are done in a team of 2 students—this is for encouraging discussions in programming analysis terminologies and languages. Final project is done in a team of 3-4 students—this is for you to experience collaboration and develop a big project within a semester. Unless specified, the grades will be the same for all the students in the team.

Submission: Homework is due Fridays 11:59pm on Canvas. Interview homework is due Fridays 3-5pm in TA office hours.

7 Academic Integrity

For the programming assignments, you are expected to write, debug and submit your own code. You should not copy the code from your classmates or from the Internet. You should not share your code, or edit and debug other people's code. For the written assignments, you should not share or copy solutions. However, you can discuss the approaches and algorithms with your classmates, TA and instructor. Once you understand and learn from the discussions, you are expected to finish the assignments independently.

Iowa State University's policy on academic dishonesty: Suspected academic misconduct will be reported to the dean of students office <http://www.dso.iastate.edu/ja/academic/misconduct.html>

8 Accommodations for Disabilities

We would like to hear from you if you have a disability that may require some modification of seating, testing, or other class requirements. If so, please request that the Disability Resources staff send a Student Academic Accommodation Notification form verifying your disability and specifying the accommodation you will need. Then bring the Accommodation Notification form along and talk to the instructor as soon as possible so appropriate arrangements may be made.

9 Free Expression

Iowa State University supports and upholds the First Amendment protection of freedom of speech and the principle of academic freedom in order to foster a learning environment where open inquiry and the vigorous debate of a diversity of ideas are encouraged. Students will not be penalized for the content or viewpoints of their speech as long as student expression in a class context is germane to the subject matter of the class and conveyed in an appropriate manner.