

# COM S 413/513 and CPR E 513: Foundations and Applications of Program Analysis

Iowa State University

Spring 2020

**Lectures:** M W F 3:10-4:00pm Morrill 1030

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

**Office Hours:** M 4:00pm-5:00pm Atanasoff 210

**TA:** Jonathan Deneke (jdeneke@iastate.edu)

**Office Hours:** Pearson 145

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

**Homework submissions and grades:** Canvas

## 1 Course Description

Algorithms and tools for automatically reasoning about code and program executions to predict software behavior. Theory and foundations related to control flow analysis, dataflow analysis, abstract interpretation and symbolic execution. Applications of program analysis to improve software security, performance and testing. 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 for COM S 413 students

After successfully completing this course, students of COM S 413 will:

1. know terminologies and background needed to read program analysis literature,
2. learn classical program analysis algorithms,
3. be able to work with open source development environment and tools such as github and docker,
4. be able to research frontier program analysis tools to address software challenges, and
5. have hands-on experiences on implementing and improving program analysis tools

## 3 Course Objectives for COM S/CPR E 513 students

After successfully completing this course, students of COM S/CPR E 513 will:

1. know terminologies and background needed to read program analysis literature,

2. learn classical program analysis algorithms,
3. be able to work with open source development environment and tools such as github and docker,
4. be able to research frontier program analysis tools to address software challenges, and
5. gain initial research experience in program analysis and software engineering

## 4 Prerequisites

Minimum of C- in COM S 342: Principles of Programming Languages

## 5 Textbooks and Resources

The course does not have a required text book. We will use lecture notes and papers to teach and learn, though the following books contain very relevant materials to program analysis:

- *Principles of Program Analysis* by Chris Hankin, Flemming Nielson, and Hanne Riis Nielson, published by Springer, ISBN 9783662038116: a theoretical, static analysis book.
- *Advanced Compiler Design and Implementation* by Steven Muchnick, published by Morgan Kaufmann, ISBN 9781558603202: it is a compiler book that covers the topics of control flow analysis, dataflow analysis, alias analysis, and the applications of program analysis in compiler optimizations.

You are also welcomed to check out the program analysis courses taught by other instructors:

- Alex Aiken, Stanford, CS 357 Techniques for Program Analysis and Verification
- Monica Lam, Stanford, CS243 Program Analysis and Optimization
- Jonathan Aldrich, CMU, 15-819 O Program Analysis
- Jens Palsberg, UCLA, CS232 Static Program Analysis
- Mayur Naik, Georgia Tech, CS6340 Software Analysis and Testing
- Stephen Chong, Harvard, CS252r Advanced Topics in Programming Languages
- Mooly Sagiv, Tel Aviv University, Program Analysis
- Evan Chang, University of Colorado Boulder, CSCI7135 Program Analysis: Theory and Practice

## 6 Tentative Topics

This course teaches what are the challenges in software engineering, and how program analysis algorithms and tools help solve these problems automatically.

**Software engineering challenges:**

1. finding bugs
2. automatic test input generation
3. debugging and repair
4. specification inference

5. analyzing software changes and versions
6. big code analysis
7. analyzing AI software

**Program analysis techniques:**

1. abstract interpretation
2. symbolic execution
3. dataflow analysis
4. control flow analysis
5. demand-driven analysis
6. reduction to graph reachability: points-to analysis and dependencies

## 7 Course Work and Evaluation

- Survey (15%)
- Homework assignments (55%)
- Project (30%)

Note that we do not grade late submissions. Please submit your homework on time. Undergraduate and graduate students will have different requirements for survey and projects. Please refer to each individual assignments for details.

## 8 Academic Integrity

Regarding 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. You may discuss the approaches and algorithms with your classmates. You also can ask questions in office hours.

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>

## 9 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.