

# COSE419: Software Verification

## Lecture 0 — Course Overview

Hakjoo Oh  
2024 Spring

# Basic Information

Instructor: Hakjoo Oh

- **Position:** Professor in CS, Korea University
- **Research Areas:** Software Engineering, Programming Languages
- **Office:** 616c, Science Library
- **Email:** hakjoo\_oh@korea.ac.kr
- **Office Hours:** by appointment

Course Website:

- Course materials will be available in Blackboard.

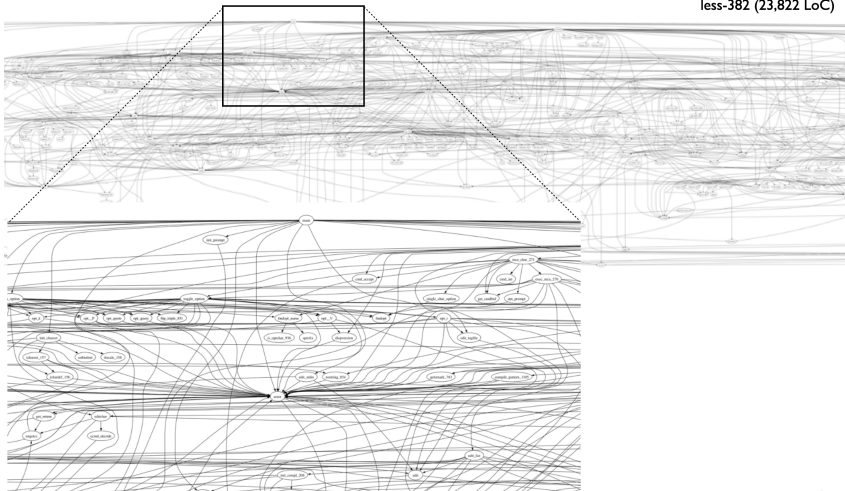
Teaching Assistants:

- Jisuk Byun (PhD candidate)
- Donguk Kim (MS candidate)

# Problem

Software is inherently complex and difficult to write, debug, and fix.

less-382 (23,822 LoC)



# Problem

- SW bugs are everywhere



Finance



Self-Driving Cars



Healthcare



Blockchain



Chip Design

- Enormous costs due to SW bugs



**606**  
software fails



**\$1.7**  
trillion



**3.6 billion**  
affected users



**268 years**  
in downtime

Software fail watch (5th edition). 2017



Software development cost

# cf) AI-based Programming



당신

Write a C program that reads a binary file containing two integers (width and height), calculates the area, and returns it.



ChatGPT

Below is a C program that reads a binary file containing two integers (width and height), calculates the area, and returns it:

```
c Copy code
#include <stdio.h>

int main() {
    FILE *file;
    int width, height;

    // Open the binary file
    file = fopen("dimensions.bin", "rb");
    if (file == NULL) {
        printf("Error opening file.");
        return 1;
    }

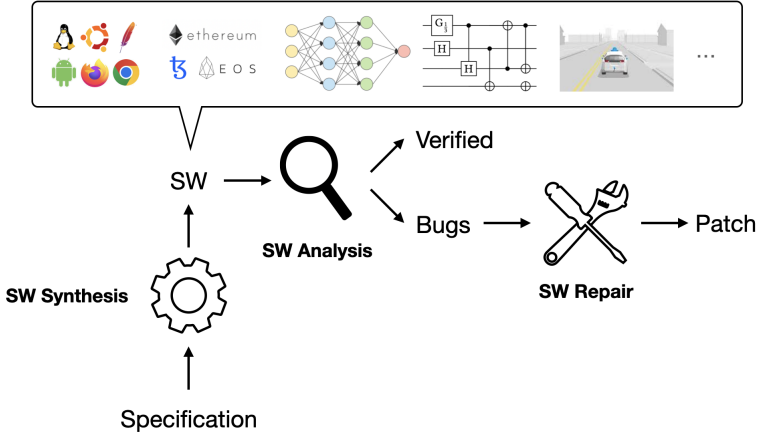
    // Read the width and height from the file
    fread(&width, sizeof(int), 1, file);
    fread(&height, sizeof(int), 1, file);

    // Close the file
    fclose(file);

    // Calculate the area
    int area = width * height;
}
```

# This course

How to correctly write, debug, and fix software?



# This course

- Goal: learn principles, techniques, and tools for correct software
  - ▶ Topics include static and dynamic analysis, symbolic execution, symbolic verification, program synthesis, program repair, etc
- Prerequisites:
  - ▶ COSE212 Programming Languages, Functional programming (OCaml)
    - ★ <https://pr1.korea.ac.kr/courses/cose212/2023/p1-book.pdf>
  - ▶ COSE215 Theory of Computation, COSE312 Compilers
- Grading (changed):
  - ▶ Attendance: 10%
  - ▶ Programming assignments: 90%
- Materials:
  - ▶ Slides will be provided
  - ▶ Reference: The Calculus of Computation. Springer

## Schedule (tentative)

Weeks	Topics
Week 1	Introduction
Week 2-3	Computational logic
Week 4-5	SMT (Satisfiability Modulo Theories) solvers
Week 6-8	Program verification
Week 9-12	Static program analysis
Week 13-14	Dynamic program analysis
Week 15	Wrap up



# Questions?