#include <iostream>

int main() {
    int i = 42;
    int j = 13;
    int k = i + j;
    std::cout << k;
    return k;
}

int main() {
    int i = 42;
    int j = 13;
    int k = i + j;
    std::cout << k;
    return k;
}
What is PhASAR?

- LLVM-based static analysis framework targeting C/C++
What is PhASAR?

- State-of-the-art static analysis infrastructure
- PhASAR caught attention from static analysis researchers and practitioners
Static Analysis: Workflow
What About the Results?
IO is Simple, Right?

- Static analysis is not one algorithm
  - Multiple data-flow solvers
  - Concrete client analysis descriptions
  - Parameterizable helper analyses (call-graph, pointer, etc.)
- Source code vs. intermediate representation (IR)
  - Non-trivial differences
- LLVM is a record-based compiler
  - Working on pointers all the time
  - Mapping needs a lot of care (especially when reading inputs)
Project Overview

- **Goal:** design and implement flexible IO capabilities that allow for
  - fast exchange of static analysis information (between different tools)
  - flexible usage of analysis tools
  - solving tasks like: “PhASAR, compute a type state analysis using this call-graph in format XYZ!”

- **Use suitable formats**
  - PhASAR’s json
  - SARIF (Static Analysis Interchange Format)
  - etc.
Project Overview

- Your tasks:
  - Get familiar with involved concepts (compilers, static analysis, formats, etc.)
  - Design input/output components
  - Develop the components in smaller teams on Github
  - Ensure usability of tooling
  - Evaluate on real-world programs
  - IDE integration using LSP (Language Server Protocol)
# Project Setting

## Requirements
- Basic C++ skills
  - (C++ course SS20, Friday 14-16/18 in C2)
- Communication skills

## Beneficial
- Knowledge of good software design and efficient programming
- Knowledge of static analysis, compilers
  - DECA course
- Knowledge of technologies such as: Git, CMake, STL, etc.

## Outcome
- Direct contribution to an open-source research project
- Deepening understanding of program analysis, programming languages and compilers
- Excellent C++ skills

Number of students: 6 - 8
Interested?
Talk to us after the presentation.
Martin.Mory@upb.de
Philipp.Schubert@upb.de