EXERCISE #23

PROGRAM INSTRUMENTATION REVIEW

Write your name and answer the following on a piece of paper

What is the difference between static and dynamic instrumentation? At what time does static instrumentation gather information?

EXERCISE #23

PROGRAM INSTRUMENTATION REVIEW

Write your name and answer the following on a piece of paper

What is the difference between static and dynamic instrumentation? At what time does static instrumentation gather information?

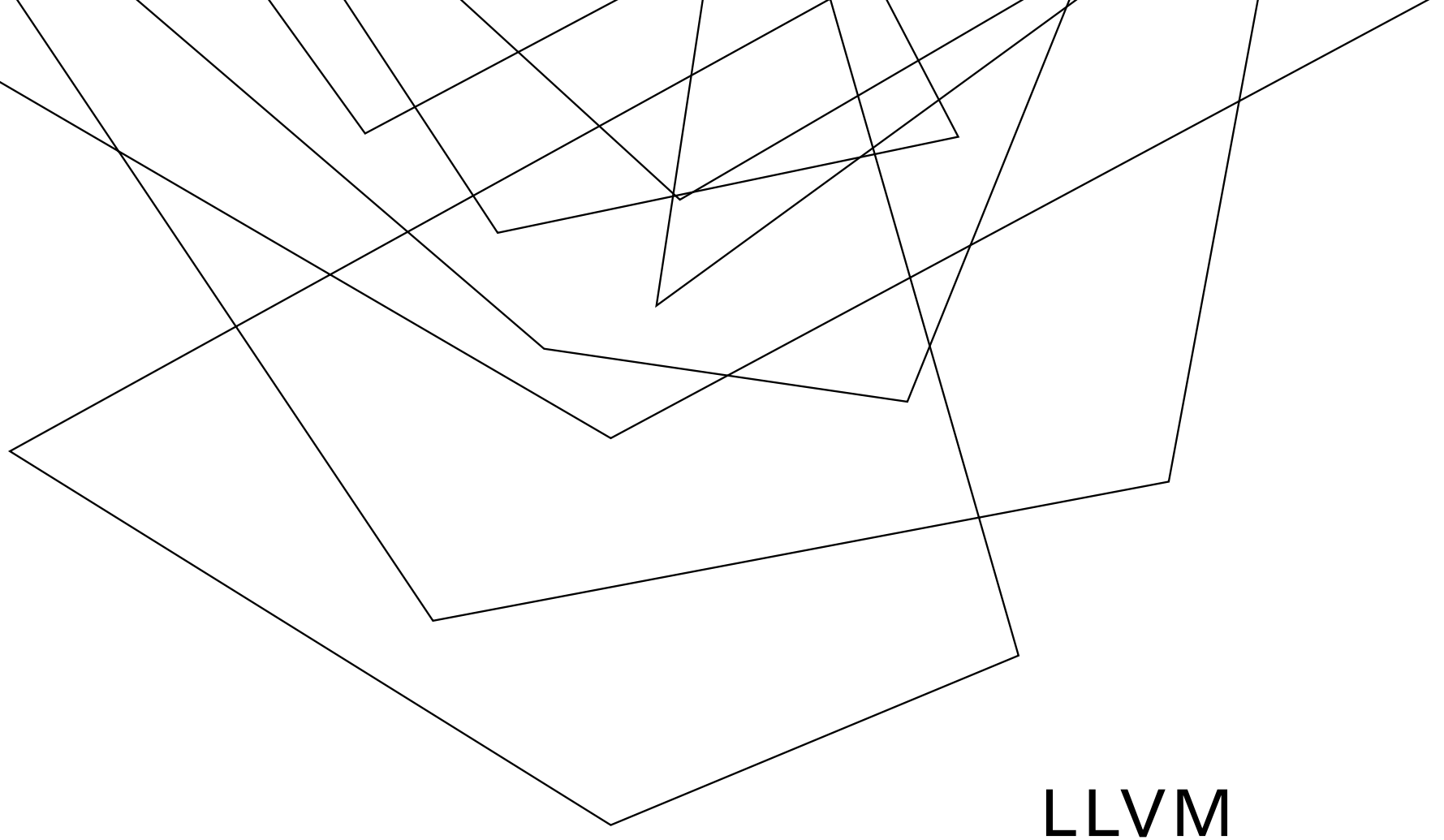
ADMINISTRIVIA AND ANNOUNCEMENTS

Quiz 2 Review Room :

LEEP2 1420, TONIGHT @ 6:30 PM

Thanks for the nom!

R2 & This weeks exercises now on Canvas



LLVM INSTRUMENTATION

EECS 677: Software Security Evaluation

Drew Davidson

PREVIOUSLY: STATIC INSTRUMENTATION

REVIEW: LAST LECTURE

BIG IDEA: REWRITE THE PROGRAM TO TATTLE ON ITSELF

BENEFITS

Cover for the false-positive problem of purely static bug-finding

PREVIOUSLY: STATIC INSTRUMENTATION

REVIEW: LAST LECTURE

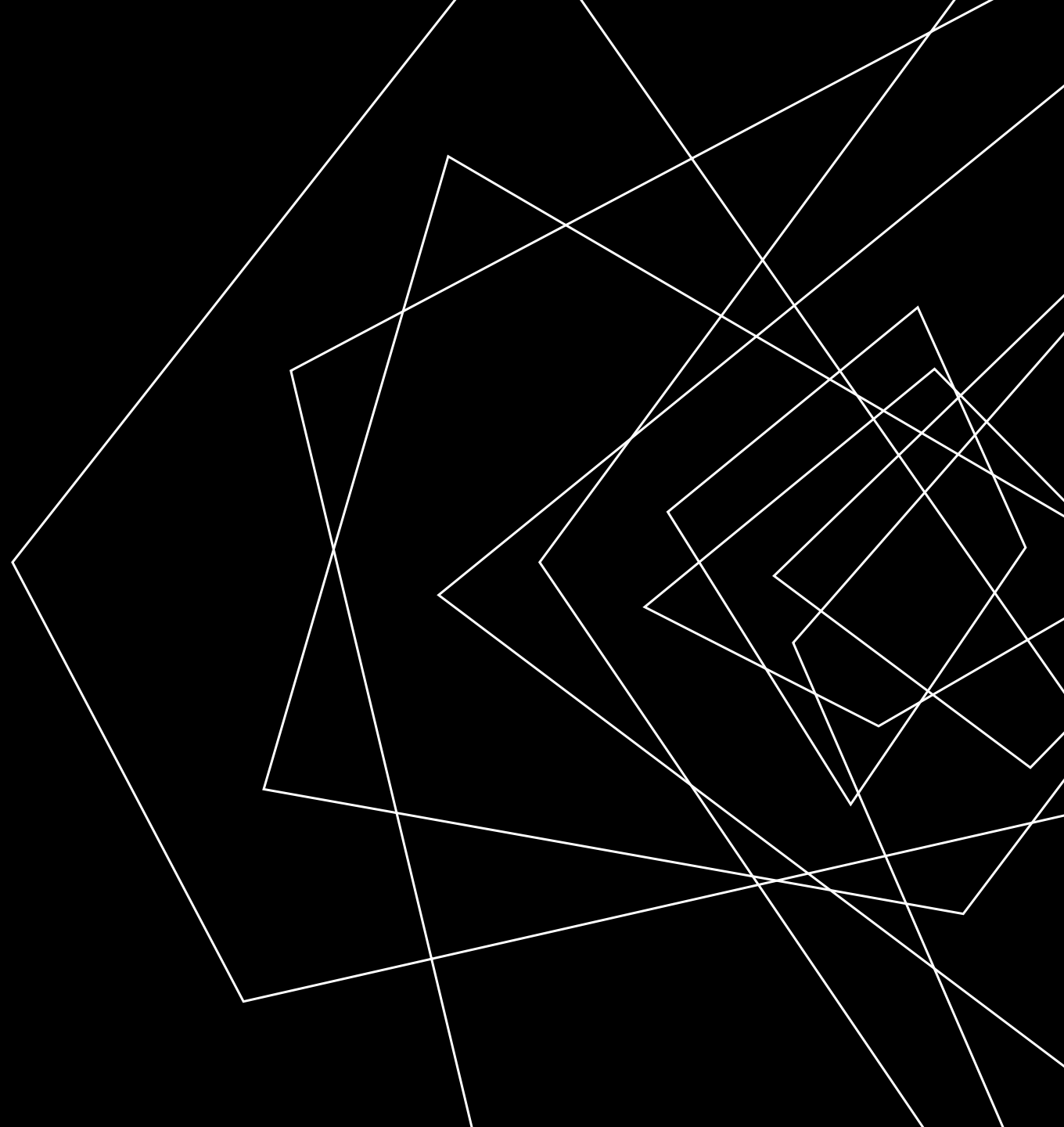
USES OF STATIC INSTRUMENTATION

Diagnostics: Figure out what the program is doing

Static refinement: Determine if a bug is real?

LECTURE OUTLINE

- Developing LLVM Instrumentation
- Example Instrumentation



CUSTOM INSTRUMENTATION

PROGRAM INSTRUMENTATION: APPROACH

The Previous Example took advantage of pre-existing instrumentation

What if we wanted to make our own custom instrumentation?

CUSTOM INSTRUMENTATION

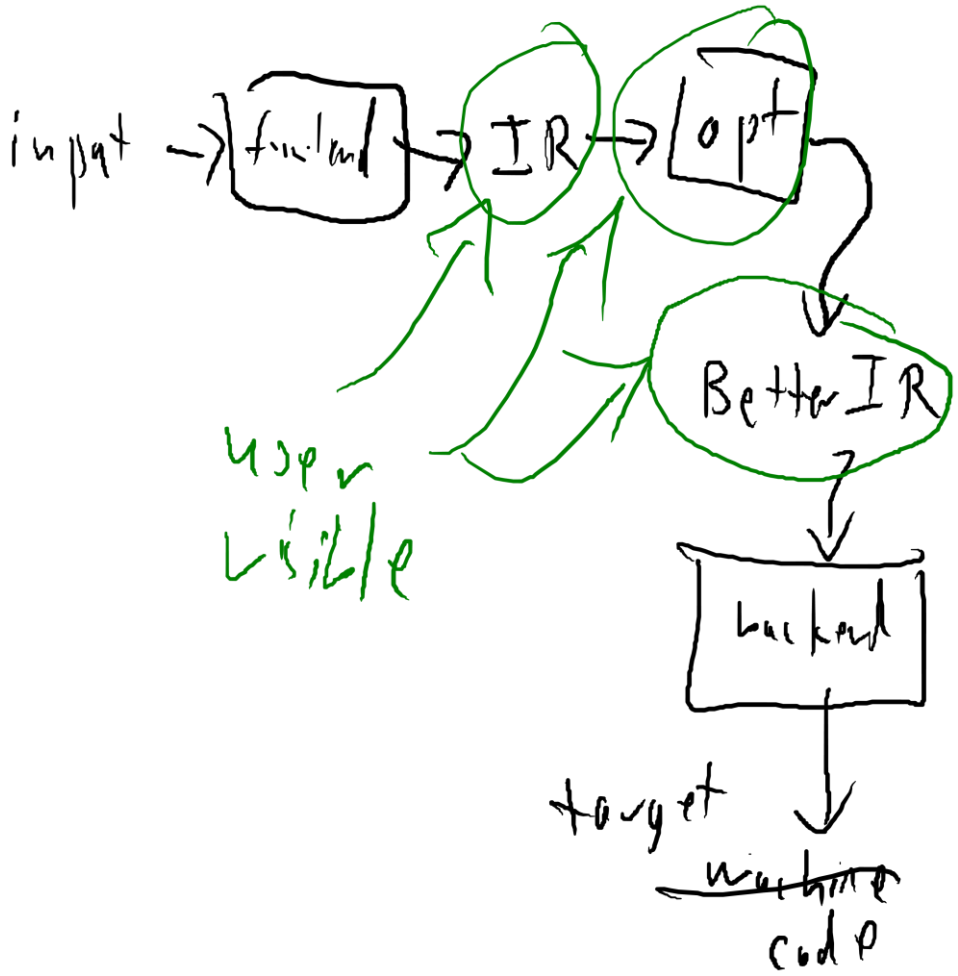
PROGRAM INSTRUMENTATION: APPROACH

GETTING STARTED

1) Reference the LLVM API
2) Build our own (trivial) analysis pass
3) Hook into the LLVM opt infrastructure
4) Run our analysis pass

GOING FURTHER

Insert more full-featured functionality
(https://llvm.org/doxygen/classllvm_1_1IRBuilder.html)



EXAMPLE: LLVM CUSTOM INSTRUMENTATION

PROGRAM INSTRUMENTATION: APPROACH

LET'S REMOVE AND ADD SOME INSTRUCTIONS!

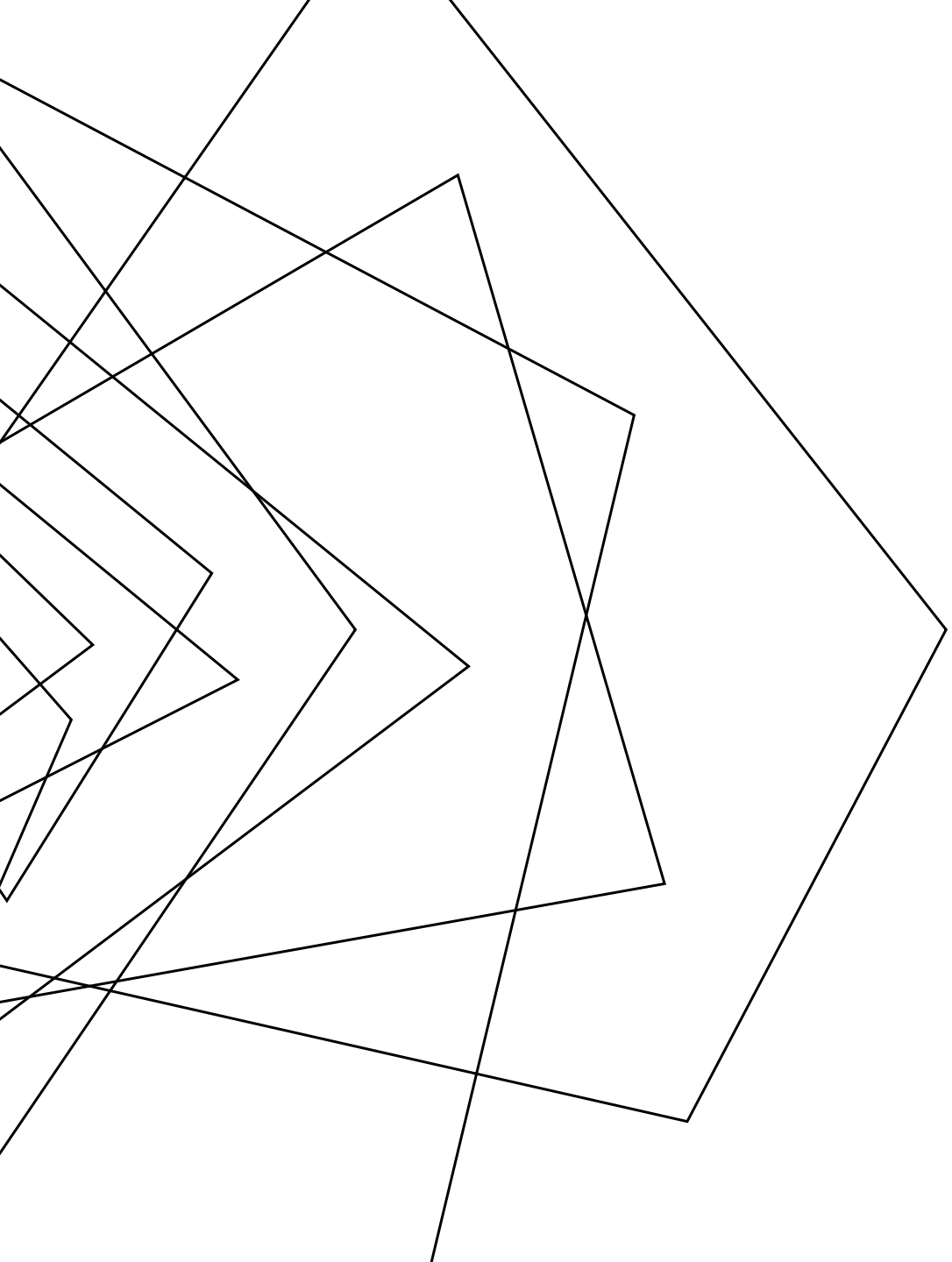
Consider a simple "add2" program:

```
#include <stdio.h>
int main(int argc, const char**
argv) {
    int num;
    scanf("%i", &num);
    printf("%i\n", num + 2);
    return 0;
}
```

EXAMPLE: LLVM CUSTOM INSTRUMENTATION

PROGRAM INSTRUMENTATION: APPROACH

Let's take it to the terminal!



WRAP-UP

WE'VE DESCRIBED THE THEORY AND PRACTICE OF PROGRAM INSTRUMENTATION

Next time: Consider how we generate test cases