

# IS593: Language-based Security

## 1. Introduction

Kihong Heo



# About Me

- Instructor: Kihong Heo (허기홍, [kihong.heo@kaist.ac.kr](mailto:kihong.heo@kaist.ac.kr))
- Research area: program analysis, SW security, programming language
- Homepage: <https://kihongheo.kaist.ac.kr>
- Office: N5 2321
- Office Hours: Mon 10:30 - 11:30

# Course Information

- Course Website: <https://github.com/KAIST-IS593/is593-2020-spring>
- TAs:
  - Changhoon Song (송창훈, [songch@kaist.ac.kr](mailto:songch@kaist.ac.kr))
  - Hyunsoo Shin (신현수, [hyunsooda@kaist.ac.kr](mailto:hyunsooda@kaist.ac.kr))
- Textbook:
  - Lecture slides will be provided
  - Xavier Rival and Kwangkeun Yi,  
[Introduction to Static Analysis: an Abstract Interpretation Perspective](#),  
MIT Press, 2020

A long time ago  
in a galaxy far, far away....

**SOFTWARE  
BUGS**

# Ariane 5 Explosion



# Software Bugs: A Persistent Problem

- A long time ago, far far away



The Patriot Missile (1991)  
Floating-point roundoff  
28 soldiers died



The Ariane-5 Rocket (1996)  
Integer Overflow  
\$100M



NASA's Mars Climate Orbiter (1999)  
Meters-Inches Miscalculation  
\$125M

- Unfortunately, it becomes your own problem now

**CNN** U.S. | World | Politics | Money | Opinion | Health | Entertainment | Tech | Style | Travel | Sports | Video | Live TV | **US**

The 'Heartbleed' security flaw that affects most of the Internet

By Heather Kelly, CNN  
Updated 5:11 PM ET, Wed April 9, 2014

A large red heart icon with a white outline, symbolizing the Heartbleed bug.

This dangerous Android security bug could let anyone hack your phone camera

By Anthony Spadafra November 23, 2019

Camera app vulnerabilities allow attackers to remotely take photos, record video and spy on users

A smartphone displaying a green binary code pattern over a keyboard background.

**AERIN MARSHALL** TRANSPORTATION 06:30 2019 07:00 AM

**What Boeing's 737 MAX Has to Do With Cars: Software**

Investigators believe faulty software contributed to two fatal crashes. A newly discovered fault will likely keep the 737 MAX grounded until the fall.

A Boeing 737 MAX airplane flying through a cloudy sky.

Homeland Security warns that certain heart devices can be hacked



New in Life & Style

Interfaith 4th-graders bond through poetry, art and Steph Curry 2:03 PM

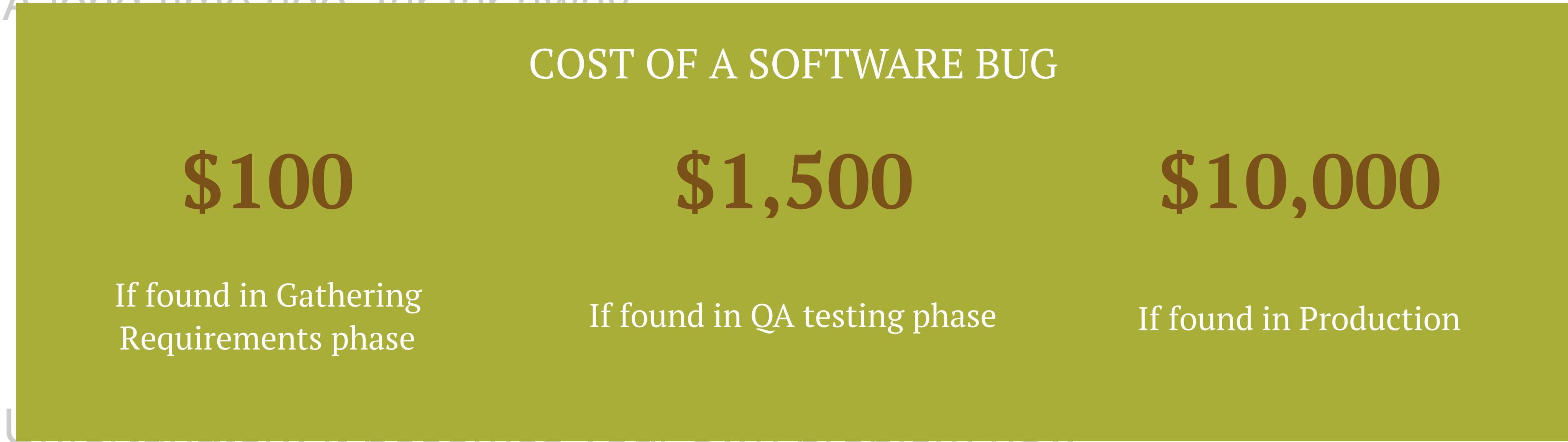
6 ways to celebrate Valentine's Day in Lake Geneva 8:55 AM

Six ways to keep your kids healthy during winter 8:56 AM

See More

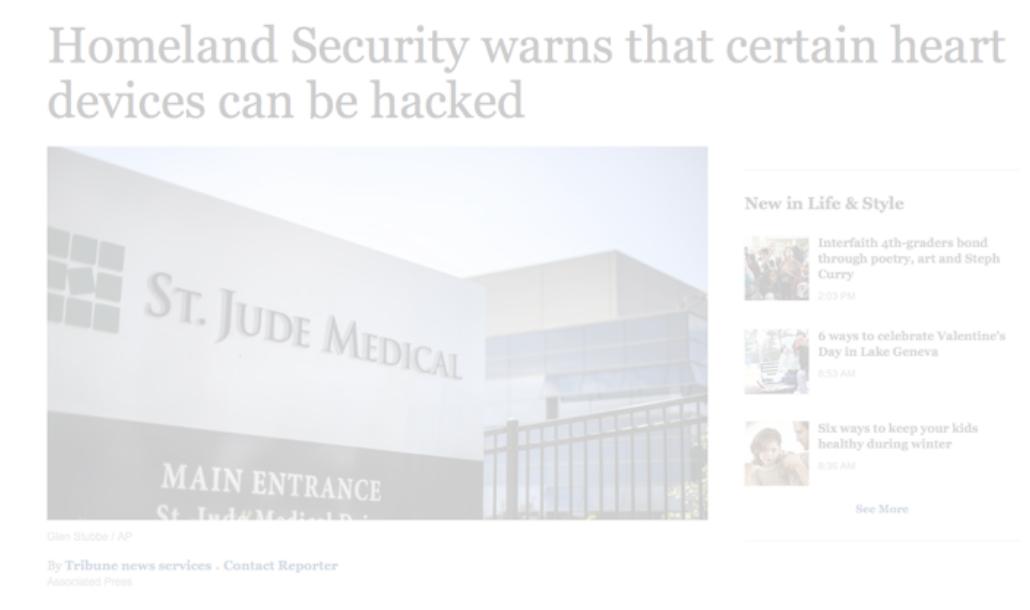
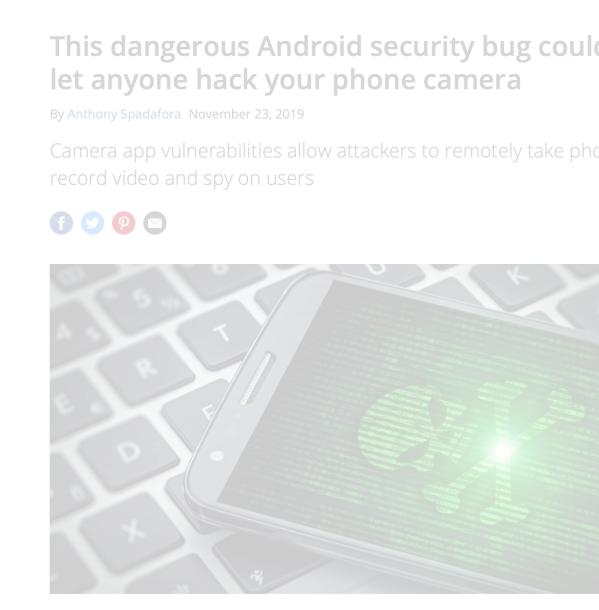
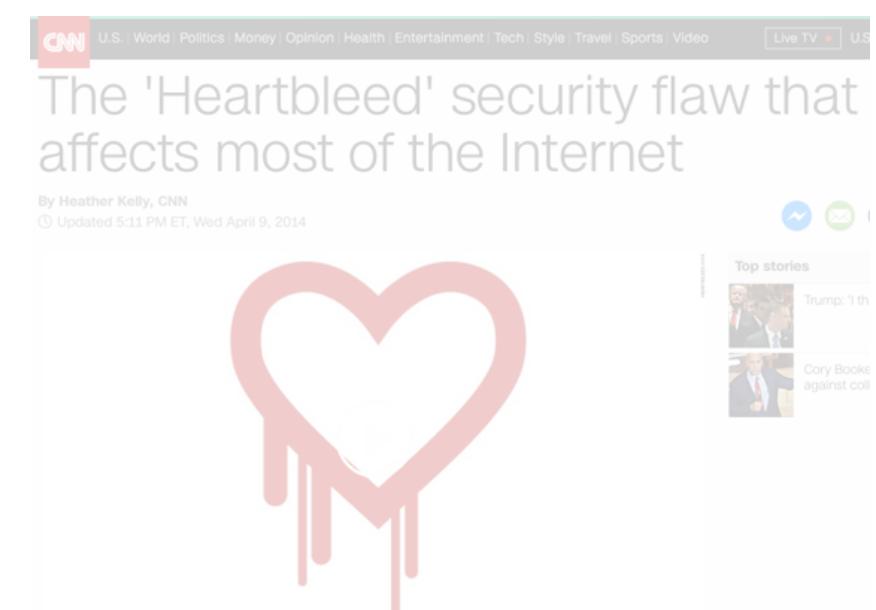
# Software Bugs: A Persistent Problem

- A long time ago, far far away...



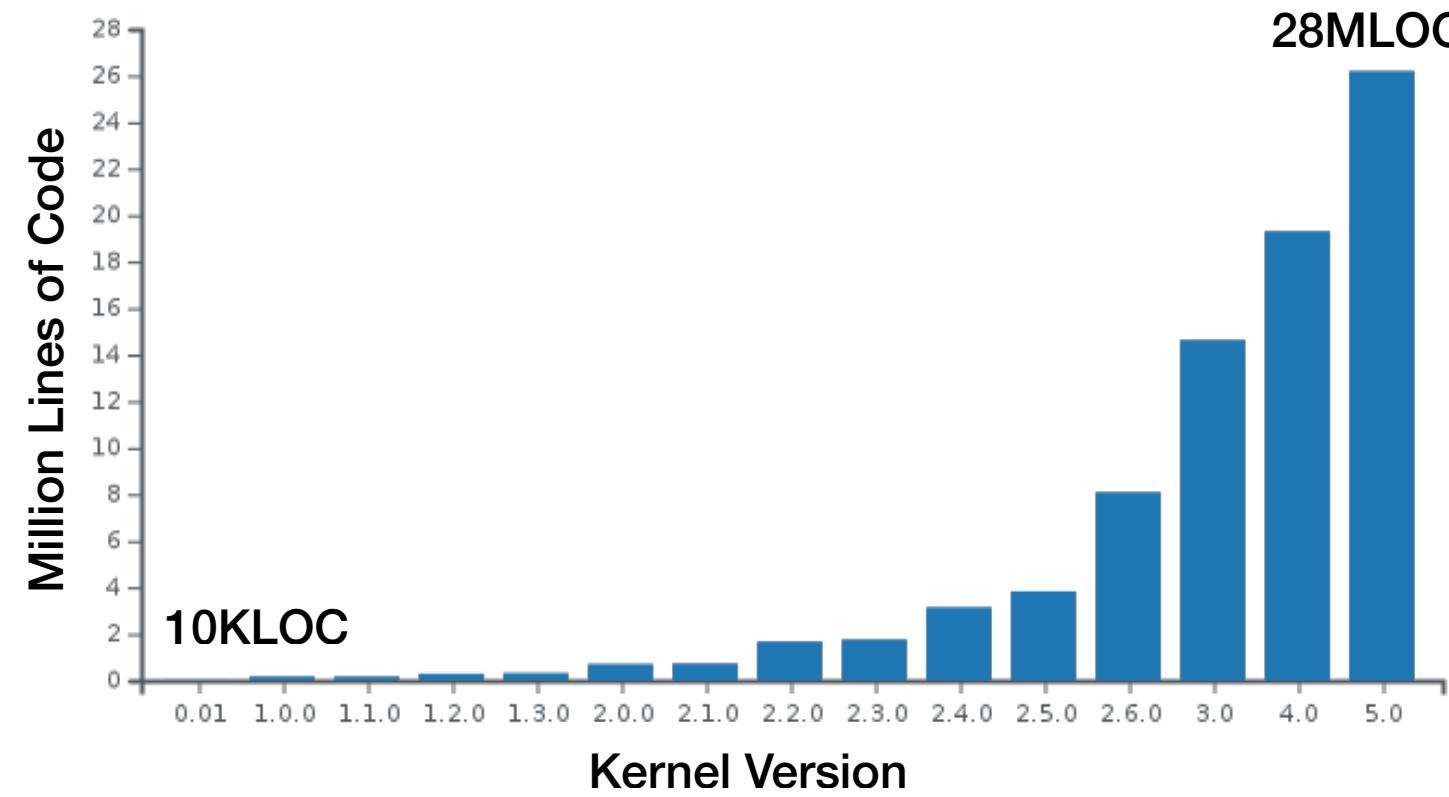
- Unfortunately, it's still a problem today...

- IBM Systems Sciences Institute, 2015

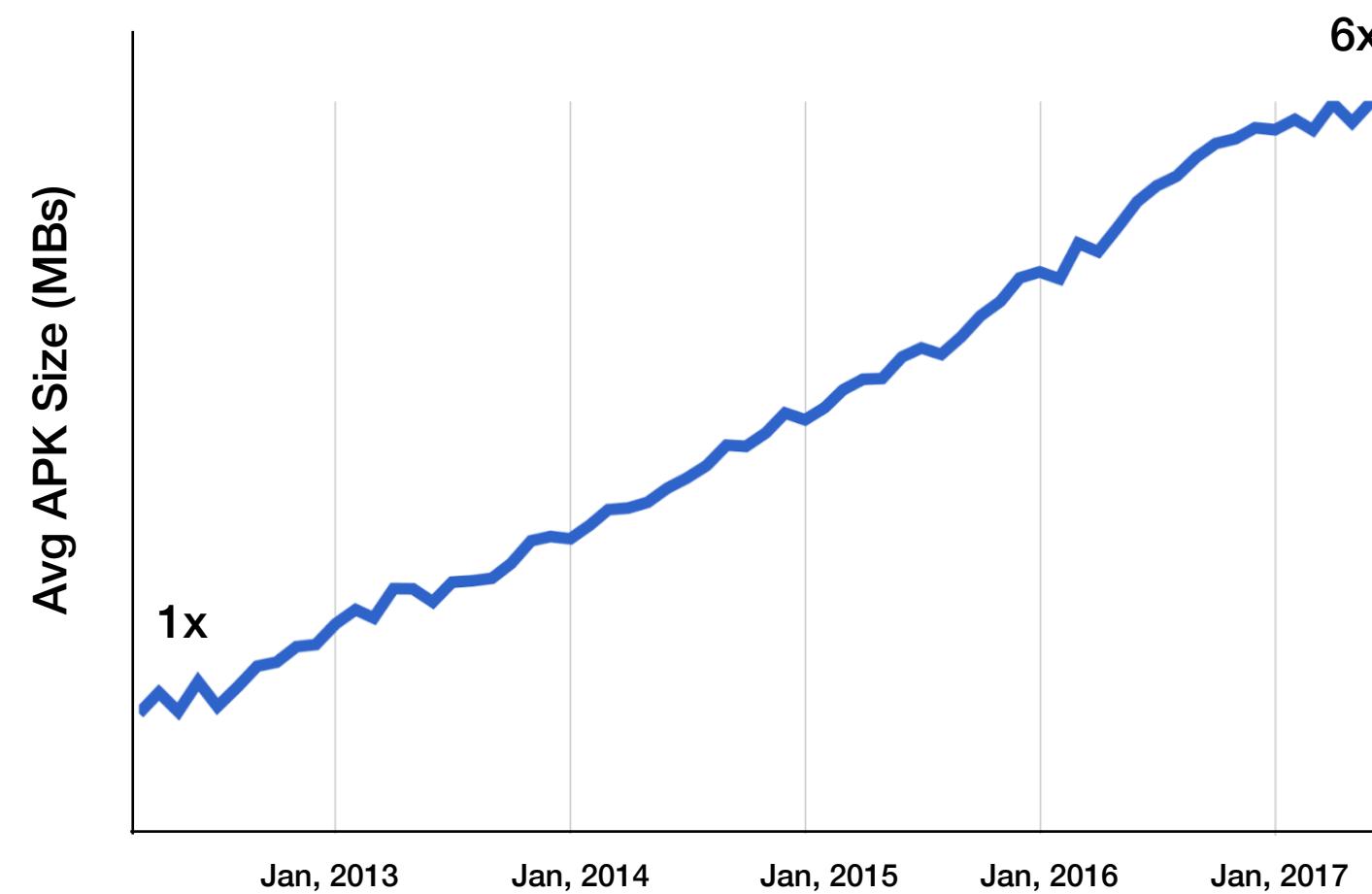


# Why Software Still Fails?

**Size of Linux Kernel**



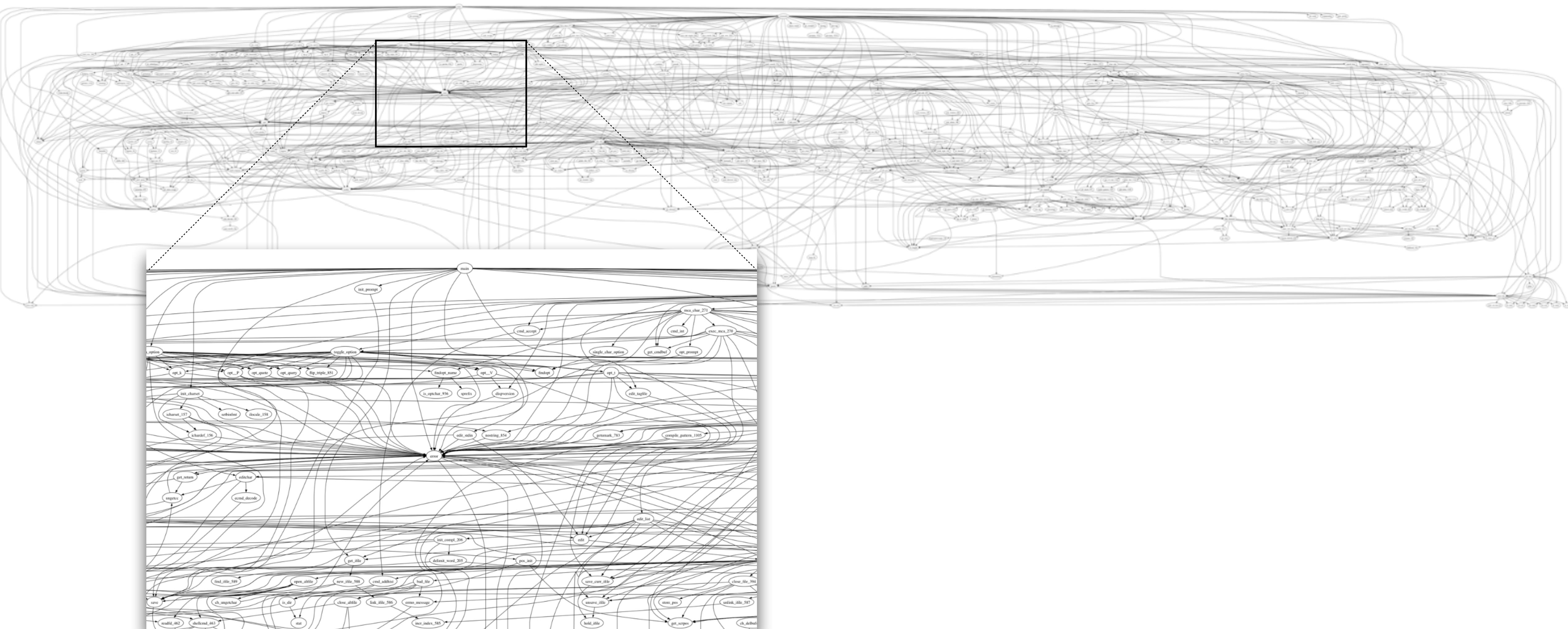
**Avg. Size of Android Apps**



**10M+ New Developers  
44M+ New Repositories  
87M+ New Pull Requests  
in 2019**

# Software Complexity

less-382 (23,822 LOC)



# Course Objectives: Principles

**Q:** How to **formally estimate** software behavior automatically before its execution?



# Course Objectives: Principles

## Artifact



## Subject



## Principle

$$\vec{F} = m\vec{a}$$



$$\nabla \cdot E = 0 \quad \nabla \times E = -\frac{1}{c} \frac{\partial H}{\partial t}$$



# Static Program Analysis

- General methodology to predict software behavior
  - **static**: before execution
  - **automatic**: software is analyzed by software (program analyzer)
  - **systematic**: foundational theory (Abstract Interpretation)
- Applications:
  - bug-finding, verification, code optimization, etc



# Success Stories

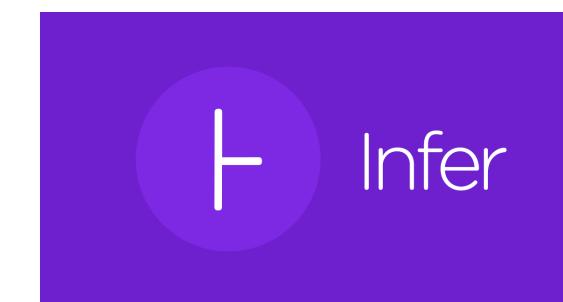
Domain-specific  
Verification



Windows Device Driver  
*Microsoft*



*Stanford / Synopsys*



*Facebook*



*SNU / Fasoo.com*



*Mathworks*



Airbus Controller  
*ENs / AbsInt*



*GrammaTech*

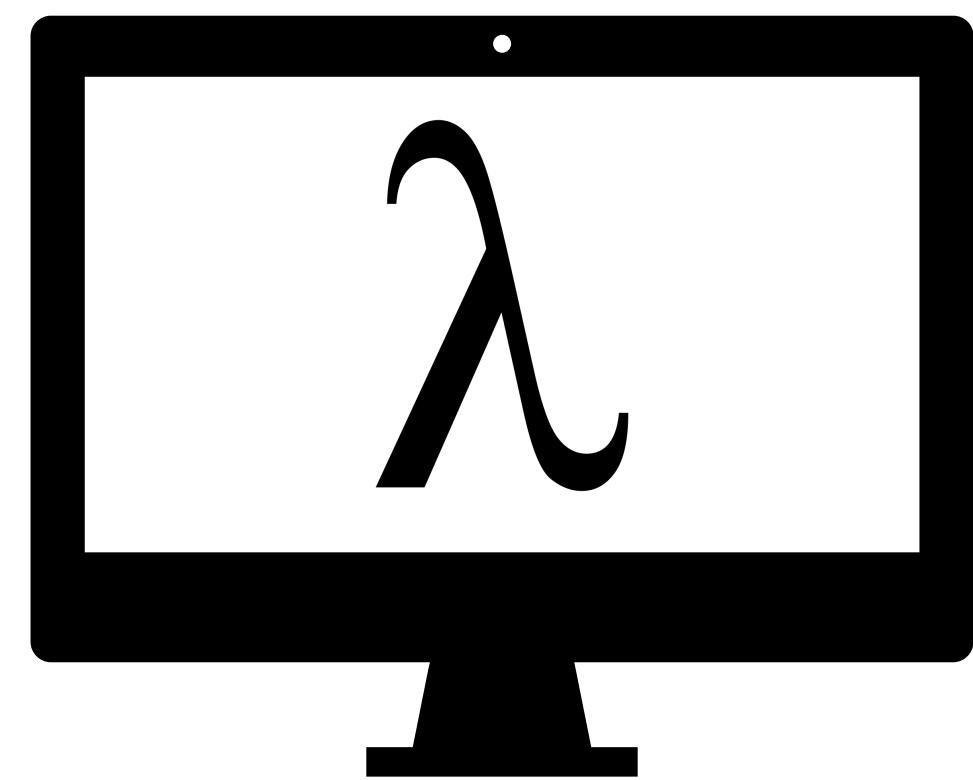


*Semmle / Github*



*JuliaSoft*

# Course Objectives: New Waves

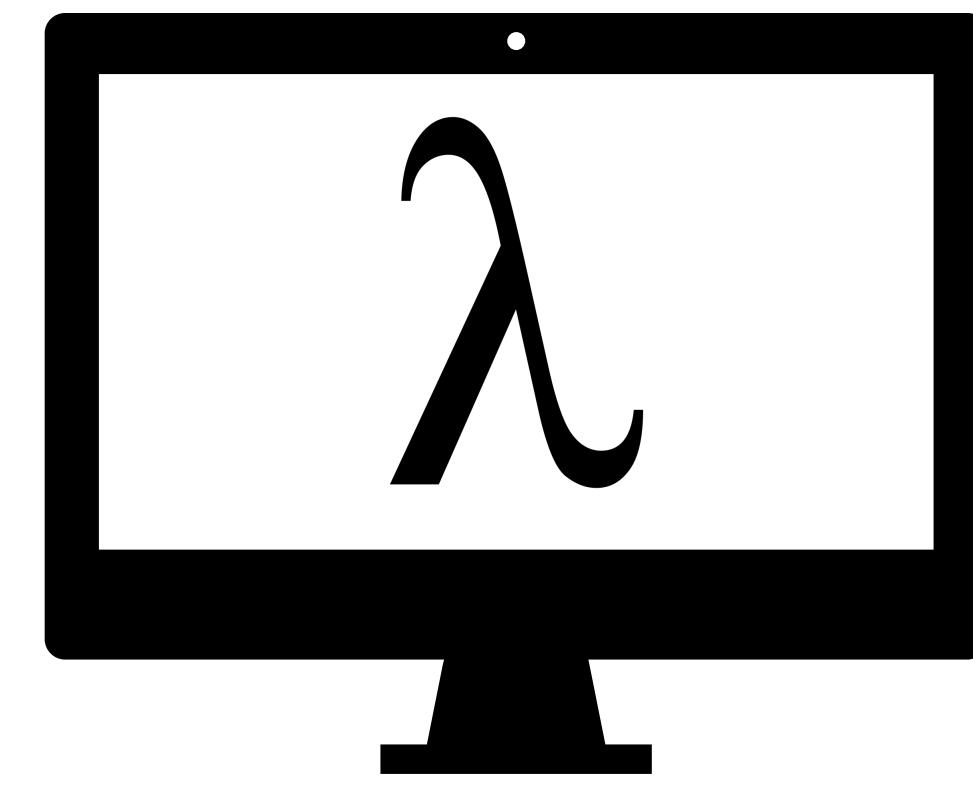


Next-generation  
Programming Systems

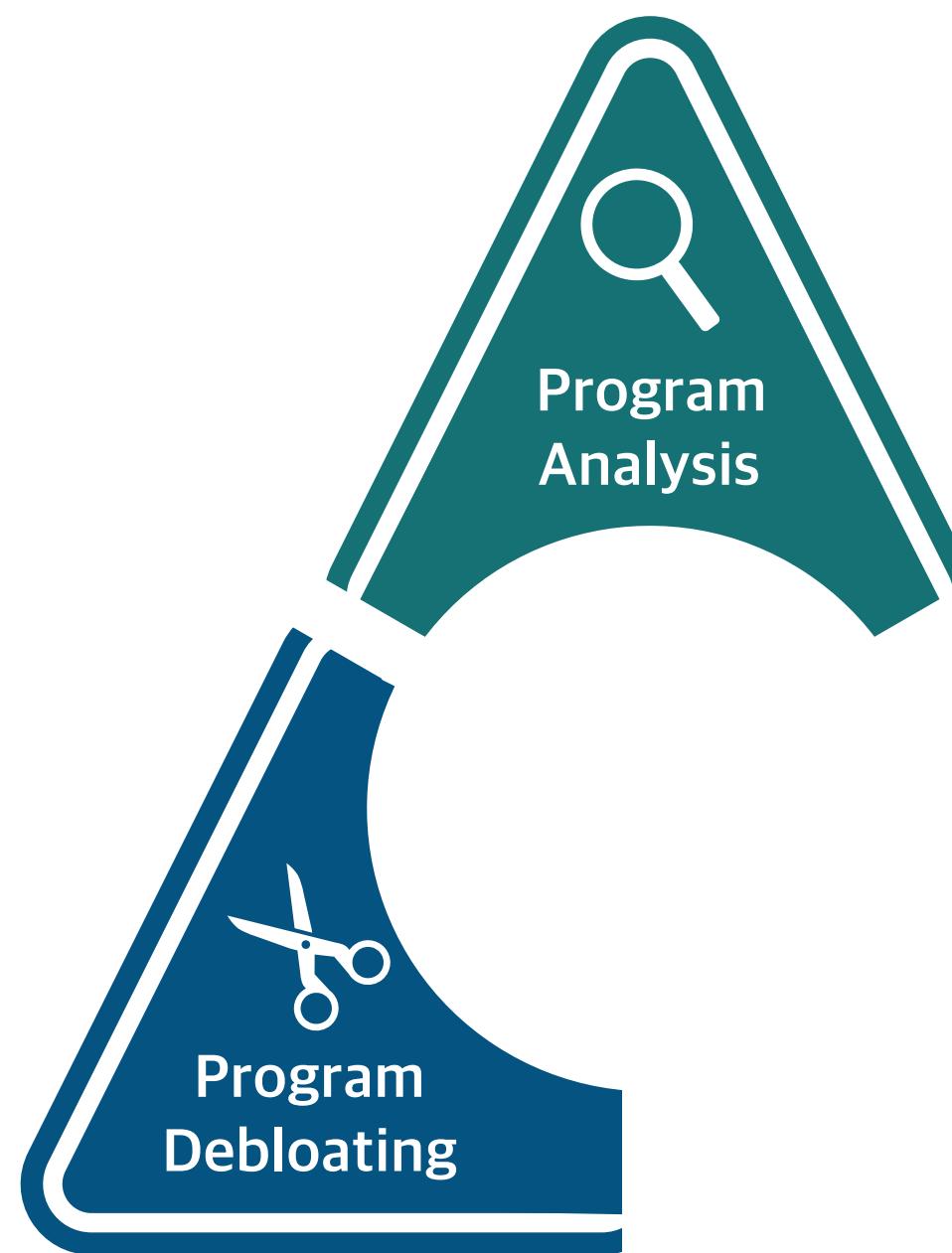
# Course Objectives: New Waves



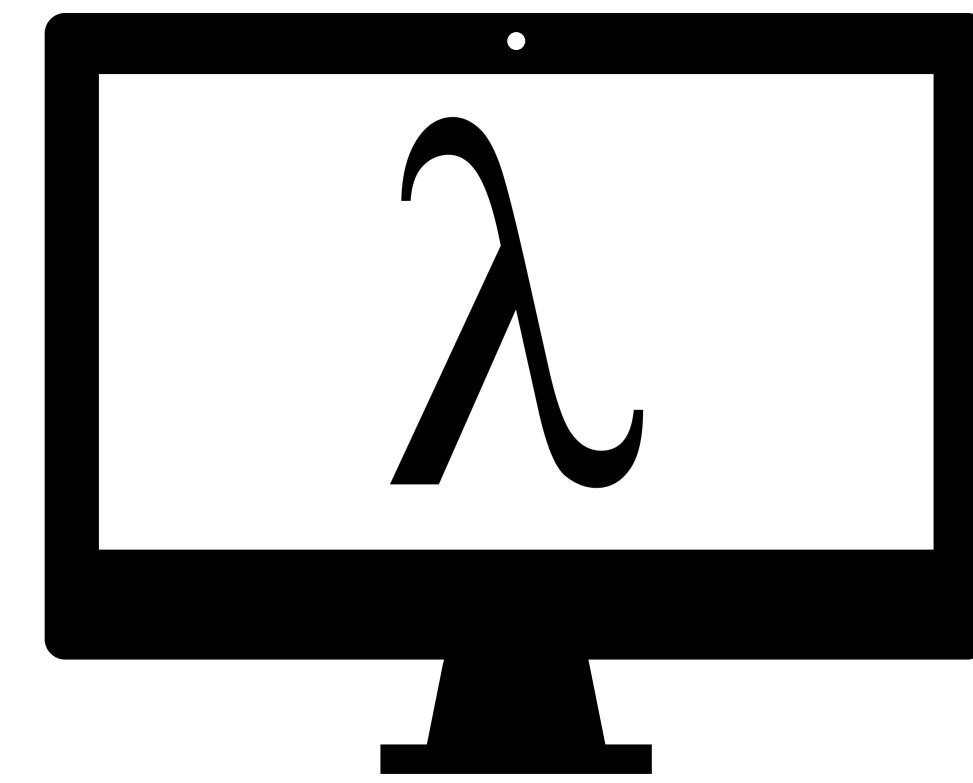
{

**Safe****Simple****Smart****Next-generation  
Programming Systems**

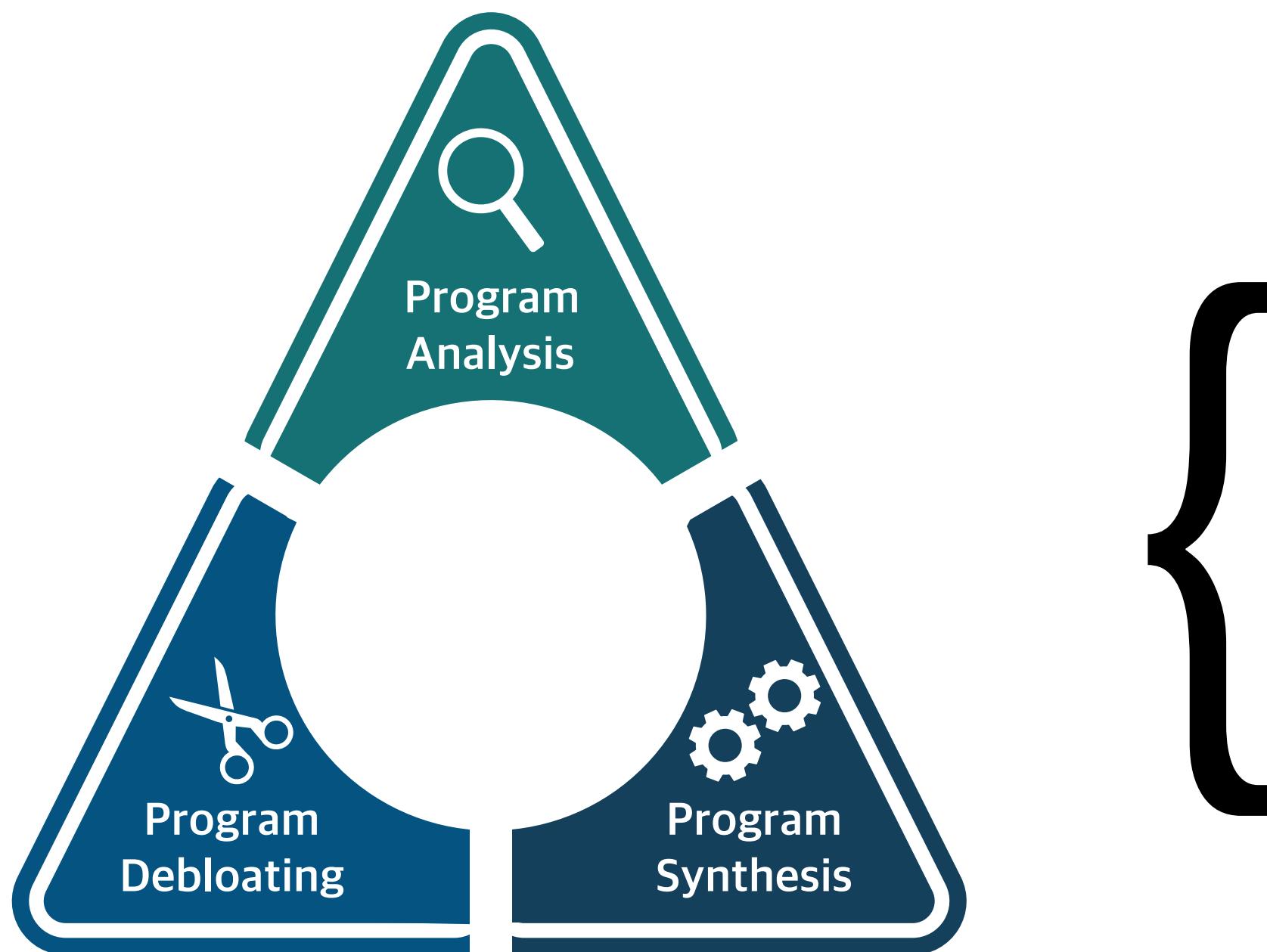
# Course Objectives: New Waves



{

**Safe****Simple****Smart****Next-generation  
Programming Systems**

# Course Objectives: New Waves



{



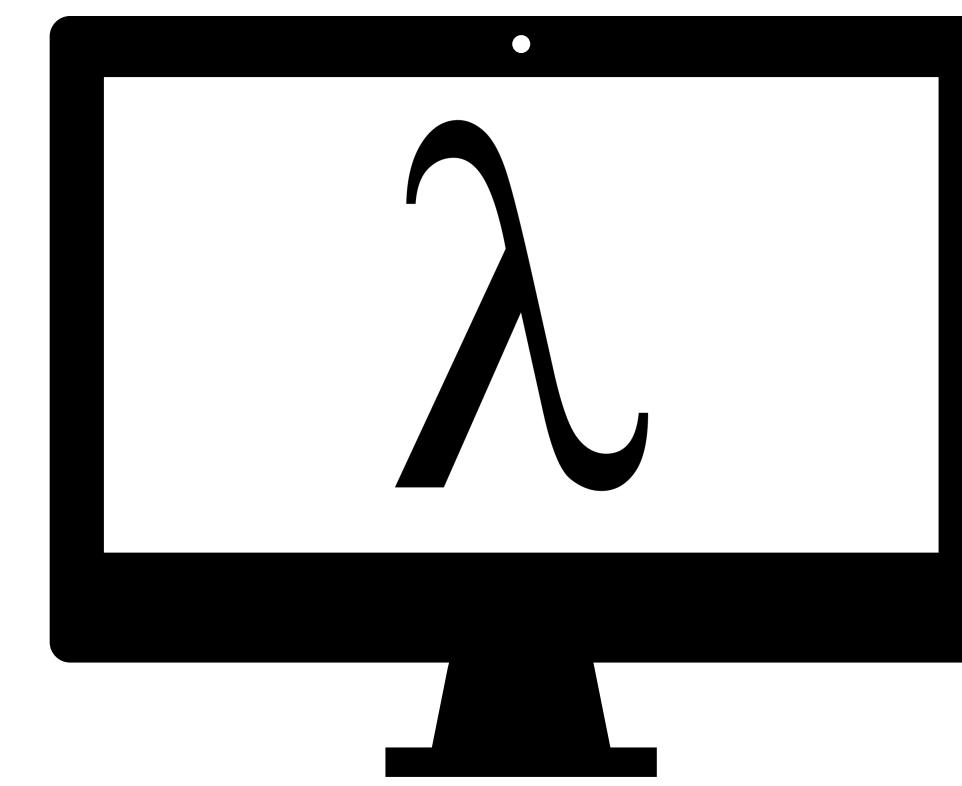
**Safe**



**Simple**

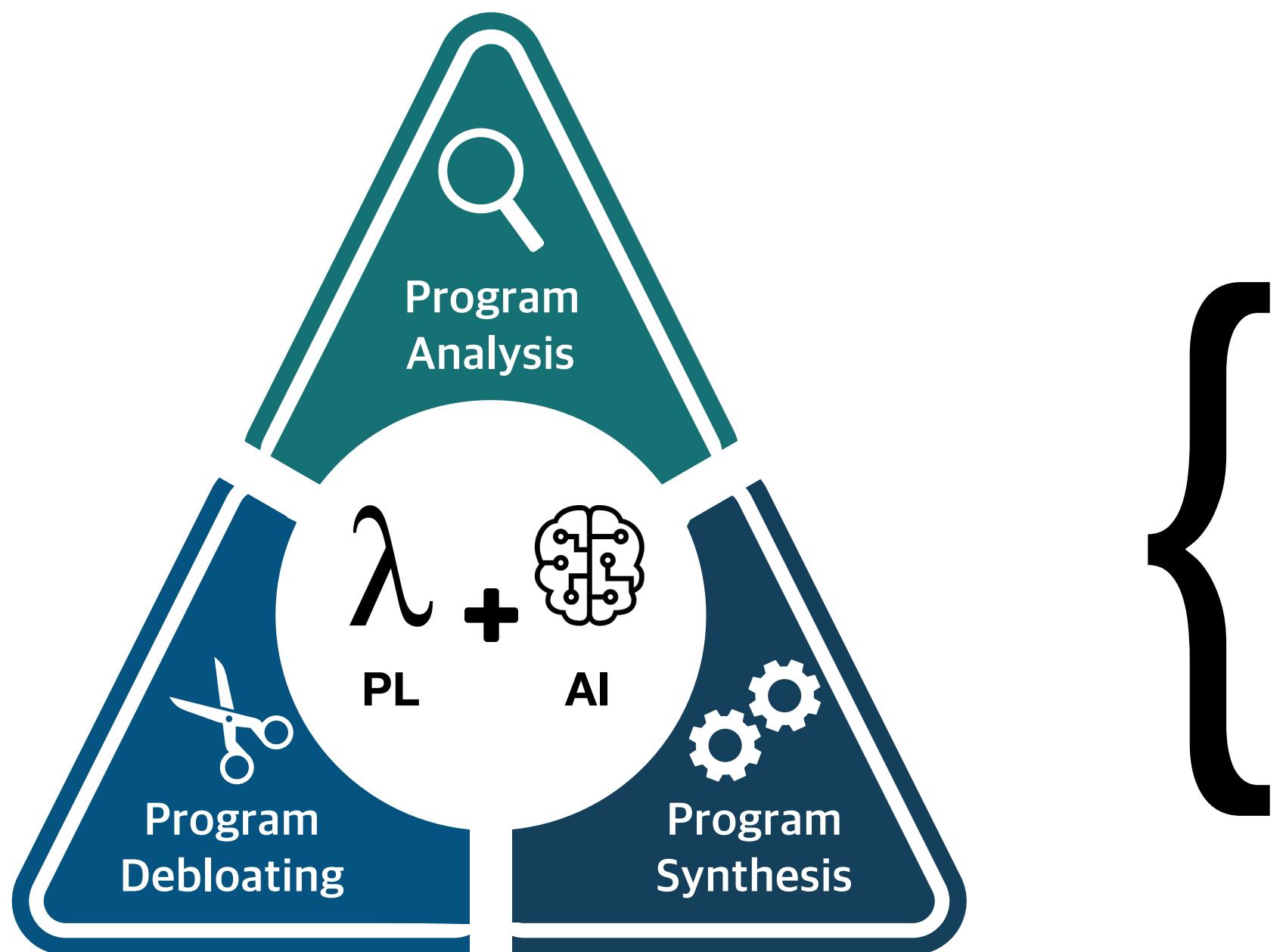


**Smart**



**Next-generation  
Programming Systems**

# Course Objectives: New Waves



{



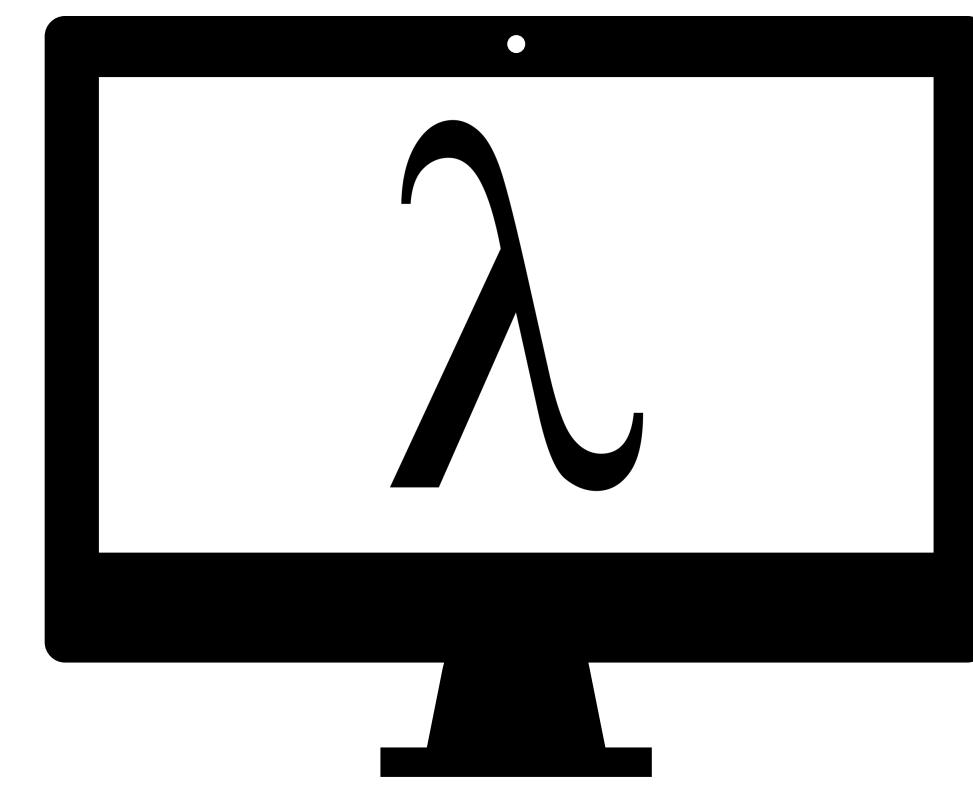
**Safe**



**Simple**



**Smart**



**Next-generation  
Programming Systems**

# Course Objectives: Practice & Challenge

- Homework: **implement** conventional program analyzers
  - 4 (main) + 1 (prerequisite) programming assignments
- Project: **design and implement** your own analyzer
  - e.g.,) your own research domain, advanced techniques, new idea

# Programming Assignments

- Programming assignments in OCaml using LLVM
  - You will write your analyzers in OCaml
  - Your analyzer will analyze LLVM IR code
- Why LLVM? (<https://llvm.org>)
- Why OCaml? (<https://ocaml.org>)



# The LLVM Compiler Infrastructure

- The de-facto standard & well-structured compiler toolchain
  - parser, code optimizer, linker, loader, debugger, etc
- A wide variety of frontends: C/C++, Obj-C, Swift, Fortran, etc
  - translated to the LLVM IR (intermediate representation)



**Apple's other open secret: the LLVM Compiler**

By Prince McLean  
Friday, June 20, 2008, 04:10 am PT (07:10 am ET)

SproutCore, profiled earlier this week, isn't the only big news spill out from the top secret WWDC conference due to Apple's embrace of open source sharing. Another future technology featured by the Mac maker last week was LLVM, the Low Level Virtual Machine compiler infrastructure project.

Like SproutCore, LLVM is neither new nor secret, but both have been hiding from attention due to a thick layer of complexity that has obscured their future potential.

**Looking for LLVM at WWDC**

Again, the trail of breadcrumbs for LLVM starts in the public WWDC schedule. On Tuesday, the session "New Compiler Technology and Future Directions" detailed the following synopsis:

Google Chrome is replacing Microsoft's C++ compiler with Clang

By Muhammad Jarir Kanji · Mar 6, 2018 14:06 EST · HOT!

A screenshot of a Windows desktop showing the Google Chrome logo. The desktop background is blue with the characteristic Windows taskbar visible at the bottom.

Alongside bringing better touch support and automatic ad-blocking for 'intrusive' ads to the desktop version of Chrome, Google is also making some changes to its browser under the hood. The company is now starting to build Chrome for Windows using the Clang compiler which it already uses for other platforms like macOS and Linux.

IBM Developer

Power developer portal Blogs Feedback

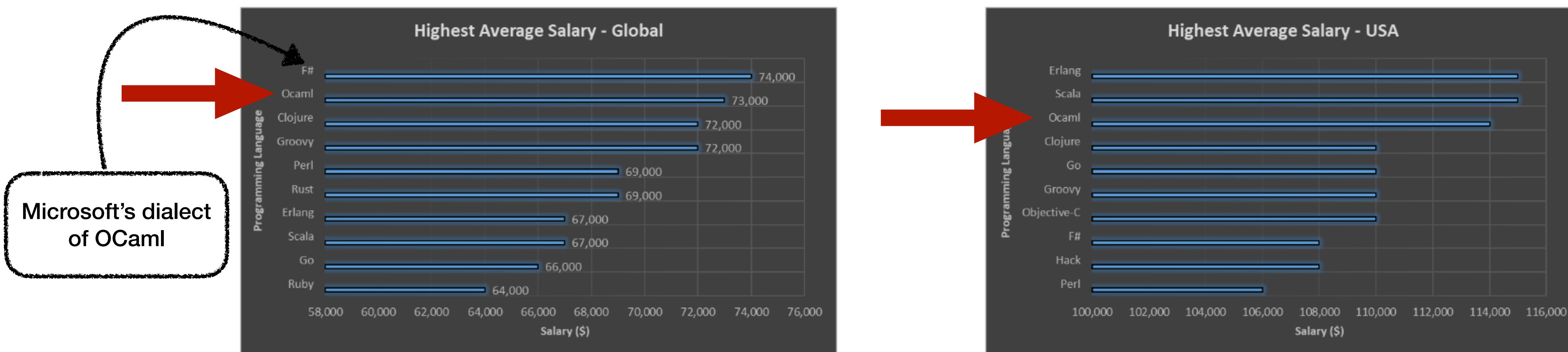
Announcements Compilers

## IBM C/C++ and Fortran compilers to adopt LLVM open source infrastructure

SiyuanZhang  
Published on February 23, 2020 / Updated on February 26, 2020

# The OCaml Language

- Simple, safe, realistic and high-level programming language
- Official OCaml bindings to LLVM API supported
- A lot of growing demands from academia and industry



StackOverflow, 2018

# Homework

- All submissions will be managed using Github / Github Classroom
  - 1. For each HW, a unique invitation URL will be posted at KLMS
  - 2. Once you accept, a private repo for your assignment will be created
  - 3. You can push as many commits as you want before the deadline
  - 4. The final commit of your master branch will be graded
    - 80% credit for 1-day late, 50% credit for 2-days late, NO credit otherwise

# Homework 0: Hello, World!

- Goal: setting up and getting familiarized with OCaml and Git environments
  - Implement your “hello-world” program in OCaml
  - Push to your Github repository
  - See the result in [travis-ci.com](https://travis-ci.com) (sign in with your Github account)
- The invitation URL is posted at KLMS
- Due on Mar 20 23:59:59

# Homework 0: Hello, World!

1. Accept the invitation  
and have your repository

The screenshot shows a GitHub repository page. At the top, it displays the repository name 'KAIST-IS593 / is593-2020-spring-hw0-KihongHeo' as private, with 1 watch, 0 stars, and 0 forks. Below the header, a blue banner on the left side contains the text '1. Accept the invitation and have your repository'. The main content area shows the repository details: 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor (KihongHeo). A green bar highlights the commit count. The commit list shows an initial commit by KihongHeo for each of the files: .gitignore, .travis.yml, README.md, dune, dune-project, hello.expected, and hello.ml. All commits were made 1 minute ago. The README.md file contains the text 'Hello, World!'. A note at the bottom states: 'The goal of this homework is to setup and get familiarized the OCaml and Github environment that will be used throughout this course. Notice that this homework is prerequisite and will not be graded.'

# Homework 0: Hello, World!

## 2. Commit your code

The screenshot shows a GitHub repository page for 'KAIST-IS593 / is593-2020-spring-hw0-KihongHeo'. The repository is private, has 1 watch, 0 stars, and 0 forks. It was generated from 'KAIST-IS593/hello-world'. The repository was created by GitHub Classroom. It contains 2 commits, 1 branch, 0 packages, 0 releases, and 1 contributor (KihongHeo). The latest commit is 'd14703b' made 4 minutes ago. The commit history includes:

File	Type	Message	Time
.gitignore	Initial commit	KihongHeo HW0 done	7 minutes ago
.travis.yml	Initial commit		7 minutes ago
README.md	Initial commit		7 minutes ago
dune	Initial commit		7 minutes ago
dune-project	Initial commit		7 minutes ago
hello.expected	Initial commit		7 minutes ago
hello.ml	HW0 done		4 minutes ago

**Hello, World!**

The goal of this homework is to setup and get familiarized the OCaml and Github environment that will be used throughout this course. Notice that this homework is prerequisite and will not be graded.

# Homework 0: Hello, World!

3. See your result

The screenshot shows a GitHub Actions build summary for the repository 'KAIST-IS593 / is593-2020-spring-hw0-KihongHeo'. The build status is 'passing'. The summary includes details about the commit (d14703b), comparison (e2941b3..d14703b), and branch (master). It also lists the environment (C, AMD64, OCAML\_VERSION=4.08.0) and provides links to 'Job log' and 'View config'.

**Job log:**

```
1 Worker information
6
7 Build system information
168
169

170 Installing SSH key from: default repository key
172 Using /home/travis/.netrc to clone repository.
173
174 $ git clone --depth=50 --branch=master https://github.com/KAIST-IS593/is593-2020-spring-hw0-KihongHeo.git KAIST-IS593/is593-2020-spring-hw0-KihongHeo
184
185
186 Setting environment variables from .travis.yml
187 $ export OCAML_VERSION=4.08.0
188
189 $ export TRAVIS_COMPILER=gcc
190 $ export CC=${CC:-gcc}
191 $ export CC_FOR_BUILD=${CC_FOR_BUILD:-gcc}
192 Setting up build cache
202
203 $ gcc --version
204 gcc (Ubuntu 7.4.0-1ubuntu1-18.04.1) 7.4.0
205 Copyright (C) 2017 Free Software Foundation, Inc.
206 This is free software; see the source for copying conditions. There is NO
207 warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
208
209 $ sudo add-apt-repository -y ppa:avsm/ppa
210 $ sudo apt update
212 $ sudo apt install -y opam
213 $ export OPAMYES=1
214 $ opam init --compiler=${OCAML_VERSION} --disable-sandboxing
215 $ eval $(opam env)
216 $ opam install dune
217 $ dune test
218 The command "dune test" exited with 0.
219 store build cache
220
221 Done. Your build exited with 0.
```