

Google Internships in Programming Languages, Compilers, Performance/Optimization, and Software Engineering (2025)

Google is looking for great interns and full-time engineers with backgrounds in programming languages, compilers, optimization, and software engineering! We accept applications and host interns year round, but most positions are for summer. Slots do fill very quickly, so please apply soon. The timeline and links to apply for this year are:

Intern Season	Intern Type	Application Dates	Posting Link
Summer 2025 STEP	STEP First and second year undergraduate students	September 30, 2024 - October 25, 2024	1st year, 2nd year US, 2nd year CA
Summer 2025	Bachelors	September 30, 2024 - November 15, 2024	US, CA
Summer 2025	Masters	September 30, 2024 - November 15, 2024	US, CA
Summer 2025	PhD	September 30, 2024 - February 28, 2025	US, CA

Interns will be exposed to Google's extensive internal developer tools and massive computing infrastructure. Some projects will have a strong emphasis on research and publication, with recent interns submitting to top tier conferences including ASPLOS, MICRO, PLDI, OOPSLA, and ICSE.

If your research is aligned with the topics below, please apply and let us know!

1. Software Performance and Optimization

- Datacenter performance analysis using hardware performance monitoring / low overhead instrumentation
- Performance optimizations for x86, ARM, RISC-V, GPUs and TPUs
- Feedback-directed optimization, autotuning and staged compilation
- Application co-location performance studies
- Profiling tools, perf_events, Linux perf tool
- Power and energy optimizations for the datacenter
- Performance data visualization
- Performance analysis and optimization of memory allocators
- Garbage collection, memory management and managed runtime optimization
- Distributed systems performance tools and optimization

2. Program Analysis

- Improving security via code sandboxing or hardening
- Source-to-source automatic differentiation
- Metaprogramming & advanced language features (for Swift and TensorFlow)
- Software correctness and race detection
- Static/dynamic program analyses & fuzzing
- Large scale static analysis
- Domain-specific languages & compilation
- Large-scale, automated refactoring
- Type inference, type checking, gradual typing
- Profile-guided automatic hardware accelerator extraction

3. Machine Learning and Data Analysis

- Machine learning applied to systems problems
- Compiler transformations and optimizations for machine learning systems
- Machine learning based filtering and clustering of compiler and tool errors
- Anomaly detection, time series analysis
- Code generation for GPUs, TPUs and image processing

4. Software Engineering and Productivity

- Software Developer Diversity and Inclusion
- Mining software repositories
- Understanding engineering productivity
- Correlation analysis of software metrics
- Optimizing silicon compiler tools

PS: We are looking for full-time hires too!