

IN THE

**United States Court of Appeals  
for the Federal Circuit**

---

ORACLE AMERICA, INC.,

*Plaintiff-Appellant,*

v.

GOOGLE LLC,

*Defendant-Cross-Appellant,*

---

Appeals from the United States District Court for the  
Northern District of California in No. 10-CV-3561, Judge William H. Alsup

---

**BRIEF OF COMPUTER SCIENTISTS  
AS AMICI CURIAE IN SUPPORT OF  
PETITION FOR REHEARING EN BANC**

---

Phillip R. Malone  
Jef Pearlman  
Juelsgaard Intellectual Property and  
Innovation Clinic  
Mills Legal Clinic at Stanford Law School  
559 Nathan Abbott Way  
Stanford, CA 94305-8610  
Telephone: 650-725-6369  
Fax: 650-723-4426

*Attorneys for Amici Curiae*

**UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT**  
**ORACLE AMERICA, INC. v. GOOGLE LLC**

Case No. 17-1118, 17-1202

**CERTIFICATE OF INTEREST**

Counsel for the:

(petitioner)  (appellant)  (respondent)  (appellee)  (amicus)  (name of party)

**Computer Scientists**

certifies the following (use "None" if applicable; use extra sheets if necessary):

1. Full Name of Party Represented by me	2. Name of Real Party in interest (Please only include any real party in interest NOT identified in Question 3) represented by me is:	3. Parent corporations and publicly held companies that own 10% or more of stock in the party
Computer Scientists (See Attachment A)	None	None

4. The names of all law firms and the partners or associates that appeared for the party or amicus now represented by me in the trial court or agency or are expected to appear in this court (**and who have not or will not enter an appearance in this case**) are:

Phillip R. Malone and Jeffrey T. Pearlman, Juelsgaard Intellectual Property and Innovation Clinic, Mills Legal Clinic at Stanford Law School

5. The title and number of any case known to counsel to be pending in this or any other court or agency that will directly affect or be directly affected by this court's decision in the pending appeal. *See* Fed. Cir. R. 47.4(a)(5) and 47.5(b). (The parties should attach continuation pages as necessary).

None

6/12/2018

Date

/s/ Phillip R. Malone

Signature of counsel

Phillip R. Malone

Printed name of counsel

Please Note: All questions must be answered

cc: \_\_\_\_\_

Reset Fields

## ATTACHMENT A

### List of COMPUTER SCIENTIST Amici Curiae

(In alphabetical order)

1. Tom Ball
2. Brian Behlendorf
3. Jon Bentley
4. Matthew Bishop
5. Joshua Bloch
6. Dan Boneh
7. Gilad Bracha
8. Eric Brewer
9. Frederick Brooks
10. Rick Cattell
11. Vinton G. Cerf
12. William Cook
13. Mark Davis
14. Jeffrey Dean
15. L Peter Deutsch
16. David L. Dill
17. Lester Earnest
18. Dawson Engler
19. Martin Fowler
20. Neal Gafter
21. Robert Harper
22. John Hennessy
23. Alan Kay
24. Brian Kernighan
25. David Klausner
26. Kin Lane
27. Ed Lazowska
28. Doug Lea
29. Bob Lee
30. Paul Menchini
31. James H. Morris
32. Peter Norvig

33. Martin Odersky
34. Tim Paterson
35. David Patterson
36. Alex Payne
37. Tim Peierls
38. Simon Phipps
39. Ronald L. Rivest
40. Curtis Schroeder
41. Robert Sedgewick
42. Mary Shaw
43. Dave Snigier
44. Alfred Z. Spector
45. Bjarne Stroustrup
46. Ivan E. Sutherland
47. Brad Templeton
48. Ken Thompson
49. Andrew Tridgell
50. Jeffrey Ullman
51. Andries van Dam
52. Guido van Rossum
53. John Villasenor
54. Jan Vitek
55. Philip Wadler
56. James H. Waldo
57. Dan Wallach
58. Peter Weinberger
59. Steve Wozniak
60. Frank Yellin

## TABLE OF CONTENTS

STATEMENT OF INTEREST .....	1
SUMMARY OF ARGUMENT .....	1
ARGUMENT .....	2
I. The panel’s rejection of the fair use finding disregards the technical realities of software APIs and directly conflicts with Ninth Circuit law.....	2
A. The panel disregarded Ninth Circuit law by minimizing the highly functional nature of APIs under the second fair use factor.....	3
B. The panel’s conclusion that Google’s API reimplementations were not transformative conflicts with the facts and with Ninth Circuit law.....	5
II. Software APIs are inherently functional and are not copyrightable.....	9
CERTIFICATE OF SERVICE .....	12
APPENDIX—LIST OF AMICI CURIAE .....	1

## TABLE OF AUTHORITIES

### Cases

<i>Campbell v. Acuff-Rose Music, Inc.</i> , 510 U.S. 569 (1994).....	8
<i>Oracle Am., Inc. v. Google Inc.</i> , No. C 10-03561 WHA, 2016 WL 3181206 (N.D. Cal. June 8, 2016), <i>rev'd and remanded sub nom. Oracle Am., Inc. v. Google LLC</i> , 886 F.3d 1179 (Fed. Cir. 2018).....	8
<i>Oracle Am., Inc. v. Google LLC</i> , 886 F.3d 1179 (Fed. Cir. 2018) [hereinafter <i>Oracle II</i> ].....	<i>passim</i>
<i>Sega Enters. v. Accolade, Inc.</i> , 977 F.2d 1510 (9th Cir. 1992), <i>as amended</i> (Jan. 6, 1993) .....	<i>passim</i>
<i>Sony Comput. Entm't, Inc. v. Connectix Corp.</i> , 203 F.3d 596 (9th Cir. 2000).....	<i>passim</i>

### Statutes

17 U.S.C. § 102(b) .....	9
--------------------------	---

### Briefs

Computer Scientists Amicus Brief I, No. 13-1021, ECF No. 118 (May 30, 2013).....	9
Computer Scientists Amicus Brief II, No. 17-1118, ECF No. 175 (May 30, 2017).....	3, 10

## STATEMENT OF INTEREST<sup>1</sup>

Amici are 60 individual computer scientists, engineers, and professors who are pioneering and influential figures in the software and computer industries.

Amici have been widely recognized for their achievements.<sup>2</sup> Amici join this brief because they believe, based on their technical knowledge and experience, that Google's reimplementation of certain Java application programming interfaces (APIs) to develop a fundamentally new and different software platform, Android, was fair use and that APIs are highly functional and not copyrightable.<sup>3</sup>

## SUMMARY OF ARGUMENT

As computer scientists, amici have spent decades creating, using, and reimplementing APIs as part of their careers inventing, building, and teaching others about computer systems. For that entire time, the industry has operated

---

<sup>1</sup> No party or party's counsel authored any part of this brief or contributed money towards its preparation or submission. No one, other than amici and their counsel, contributed money towards the preparation or submission of this brief. This brief is being tendered with a motion for leave to file. Google has consented to the filing of this brief; Oracle has stated that it objected to the filing.

<sup>2</sup> Amici's biographies are attached as Appendix A. Amici sign this brief on their own individual behalf. A small number of the amici, indicated by (\*) next to their names, are presently Google employees, consultants, and/or directors, and four amici, indicated by (†), were retained as experts by Google or testified as fact witnesses in this case. Each of these amici sign this brief as individual computer scientists whose work in the field long preceded their affiliation with Google or their participation in this case, not on behalf of Google or at Google's request.

<sup>3</sup> Many of the amici here previously filed amicus briefs in this case, one in the first appeal explaining why APIs are not copyrightable and another in the case below describing why Google's reimplementation of Java API declarations was fair use.

under the shared understanding that APIs, being functional, are free to reuse.

Amici, and virtually the entire computer software community, have relied on API reimplementations and the programs built on them to create and operate new software and hardware, and they depend on APIs remaining open to sustain widespread compatibility standards used by startups and incumbents alike. The decisions by the panel below vitiate this fundamental understanding.

The decisions swept aside the jury and district court's fair use and copyrightability determinations that were based on a highly technical factual record. In so doing, the panel fundamentally mischaracterized the technical nature of API technology and created clear conflict with the Ninth Circuit by disregarding its law governing the assessment of fair use in copying software. These fair use and copyrightability determinations are of exceptional importance; if not revisited, they will dangerously undermine the settled expectations not only of computer scientists but also of the entire computer industry that rely upon the open nature of APIs.

## ARGUMENT

### **I. The panel's rejection of the fair use finding disregards the technical realities of software APIs and directly conflicts with Ninth Circuit law.**

The panel correctly recognized that it must apply Ninth Circuit law, *Oracle Am., Inc. v. Google LLC*, 886 F.3d 1179, 1190 (Fed. Cir. 2018), but then disregarded key Ninth Circuit precedent—in particular, the two directly relevant Ninth Circuit cases that analyze fair use in the context of software interfaces, *Sega*

and *Sony*. See *Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992), *as amended* (Jan. 6, 1993); *Sony Comput. Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000). The panel cites them merely to recognize that the “Ninth Circuit has made it clear that some such uses can be fair” and to baldly state that “the facts relating to the copying at issue here ... differ materially from those at issue in *Sony* and *Sega*.” *Oracle II* at 1210. *Sega* is not cited again, and *Sony* only once more, in a misreading of its holding on transformativeness. See *infra* at 6.

- A. The panel disregarded Ninth Circuit law by minimizing the highly functional nature of APIs under the second fair use factor.

The panel’s approach to the second fair use factor is contrary to the technical realities of APIs and thus to Ninth Circuit law. The panel “assumed” that the jury “concluded that functional considerations were both substantial and important.” *Oracle II* at 1205. They were; as amici explained in their previous brief, APIs serve a predominantly functional purpose. Computer Scientists Am. Br. II at 25, 23-27. Whatever creativity an API may exhibit, this purpose remains, and the purpose of reimplementing an API as Google did, is to reuse that functionality.

But the panel then proceeded to minimize that importance of the second fair use factor, in direct conflict with the *Sega* and *Sony* precedents.<sup>4</sup> The panel ignored

---

<sup>4</sup> *Sega* and *Sony* each involved a defendant who copied the *entire* object code of a product, *including* all API-implementing code: In *Sega*, the defendant copied the software from three video game cartridges; in *Sony*, the defendant copied the entire

the admonishment in *Sega*, a case about reverse engineering a software API, that the “second factor . . . is important to the resolution of cases such as the one before us.” 977 F.2d at 1522 (criticizing the court for ignoring the second factor); *id.* at 1524 (“[c]omputer programs pose unique problems for the application of the ‘idea/expression distinction’ that determines the extent of copyright protection”), and in *Sony*, where, in the context of copying to reverse engineer APIs, the circuit held that the second factor “strongly favors” fair use. *Sony* at 605.

Instead, the panel relied solely on a Ninth Circuit case about fictional children’s books and several out-of-circuit cases about books, television, and toy dolls, to say the opposite—that “this second factor typically has not been terribly significant in the overall fair use balancing,” *Oracle II* at 1205 (quoting *Dr. Seuss Enters., L.P. v. Penguin Books USA, Inc.*, 109 F.3d 1394, 1402 (9th Cir. 1997)). The panel acknowledged that “the jury’s assumed view of the nature of the copyrighted work weighs in favor of finding fair use” but nevertheless erroneously accorded that factor “less significance to the overall analysis.” *Id.* at 1205.

It is highly significant. The functional aspects of an API are their core

---

Basic Input Output System (“BIOS”) of a video game console—essentially the API implementation for the whole system. In each case, the defendant was attempting to reconstruct the game system’s API—in *Sega*, to make compatible games, and in *Sony* to reimplement the API to make a compatible, software-based game system. In each case, the court held that the copying was fair use as a matter of law.

features, while the expression, even when creative, is secondary at most. Likewise, the purpose of reimplementing an API is not to reuse the expression, but for the functionality it offers in allowing programmers to communicate with code and code to communicate with code. The API declarations<sup>5</sup> at issue here are not seen by those in the software industry as independent “works.” Instead, they function as part of the Java language, which is undisputedly free to reuse. To a programmer, a strong identifying feature of any programming language is the set of API methods available. Both technically and practically, the core APIs of a language are inseparable from the language, which is purely functional. For copyright purposes, APIs represent almost exclusively “the functional aspects of a software program,” for which “the fair use doctrine preserves public access.” *See Sony* at 603 (quoting *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 432 (1984)).

B. The panel’s conclusion that Google’s API reimplementation was not transformative conflicts with the facts and with Ninth Circuit law.

The panel similarly disregarded Ninth Circuit authority on the key question of whether the reuse of software code and APIs is transformative under the first fair use factor. Google’s reimplementation of the Java API declarations in its new

---

<sup>5</sup> Throughout this brief, amici use only the term “declarations” instead of “declaring code,” which was sometimes used in the panel opinion. “Declaring code” is not a term of art, and is not used in the industry, in part because declarations *are not code*: they cannot be executed by a computer and their only function is to dictate how code communicates with other code.

Android platform is the kind of transformative use the Ninth Circuit recognized in *Sony*.<sup>6</sup> Yet the panel fundamentally misapplies *Sony*, effectively ignoring their analysis of transformativeness of highly functional APIs and software. The panel instead relies on a variety of largely inapposite cases dealing with plainly expressive works, not software.<sup>7</sup>

The panel noted that the Ninth Circuit in *Sony* had held the defendant’s use of Sony’s code—copying an entire system temporarily to reimplement the API and build a compatible replacement for Sony’s system that played existing games—to be “modestly transformative.” *Oracle II* at 1200. But the panel then concluded that Google’s use in this case—to reimplement the API and build on it to design an entirely new mobile operating system that Java developers would be able to use but would do something *new and better*—was less transformative. *Id.* at 1200.

To reach its conclusion, the panel disregarded the functional characteristics of software APIs and instead treated this case as analogous to cases like *Infinity Broad. Corp. v. Kirkwood*, 150 F.3d 108 n.2 (2d Cir. 1998). *See Oracle II* at 1202. But the Ninth Circuit previously noted the inapplicability of *Infinity* to the fair use

---

<sup>6</sup> Though *Sega* pre-dates “transformativeness” law, the *Sega* court found the first factor favored the defendant despite the use’s commercial nature. *Sega* at 1523.

<sup>7</sup> See, e.g., *Oracle II* at 1201 (quoting *Seltzer v. Green Day*, 725 F.3d 1170, 1177 (9th Cir. 2013) (addressing use of an artist’s illustration—an entirely expressive work—in a backdrop and relying on prior cases involving use of celebrity wedding photos, clips of Elvis Presley, excerpts of video footage, etc.)).

of software, distinguishing the use made in *Infinity*—“merely taking copyrighted radio transmissions and retransmitting them over telephone lines,” *Sony* at 607—from the fundamentally different copying of code by the defendant. The district court had found that the defendant’s product was not transformative because “a computer screen and a television screen are interchangeable, and the [defendant’s] product therefore merely ‘supplants’ the Sony PlayStation console.” *Id.* at 607. The Ninth Circuit rejected this as clear error. It instead found that the copying resulted in a “new platform” that allowed users to play PS games in “new environments”—personal computers—rather than the previous environment of the PS console and television. *Id.* a 606. It created a “wholly new product notwithstanding the similarity of uses and functions between it and the Sony PlayStation.” *Id.* at 606.

The panel in this case made the same mistakes the Ninth Circuit corrected in *Sony*. Android does not merely supplant Java SE by moving from a desktop format or medium to the smartphone environment. Rather, Google used the Java API declarations to create a “wholly new product,” Android, that brought some “similar uses and functions,” plus much more, to the new smartphone environment. Android created an entirely new platform for mobile devices that selectively incorporates and augments the Java API declarations to achieve compatibility.

Google reused only a small portion of the declarations in the Java SE API packages and then reimplemented them with all its own implementing code,

“adapted to the constrained operating environment of mobile smartphone devices,” *Oracle v. Google*, 2016 WL 3181206, at \*9, including unique phone features like touchscreens, GPS, accelerometers, smaller batteries, etc. Google then added many *new* APIs to form a wholly new platform that transcended the original. The district court properly concluded that a reasonable jury could have found that Google’s reimplementations and original creations, “all constituted a fresh context giving new expression, meaning, or message to the duplicated code.” *Id.* at \*9.

The panel compounded its error by examining the transformativeness only of the reimplemented APIs themselves, rather than the resulting new work (Android) as a whole, and by finding no transformativeness because the APIs serve the same purpose in Android as they did in Java SE. *Oracle II* at 1200-1201. But in *Sony* the Ninth Circuit, assessing Connectix's copying and use of the Sony BIOS to develop its own software, examined *not only* the functionality copied and used to make Connectix’s new platform, but rather the transformativeness of that work *as a whole*, concluding that “the Virtual Game Station itself is a wholly new product” that “does not merely supplant the PlayStation.” *Sony* at 607.<sup>8</sup>

The panel’s decision failed to account for the unique aspects of software when assessing transformativeness and ignored the fact that software’s functional

---

<sup>8</sup> *See Campbell v Acuff-Rose Music, Inc.*, 510 U.S. 569, 579, 581-82 (1994) (assessing transformativeness of “the new work,” 2 Live Crew’s *Pretty Woman*, not merely the few lines copied from the original song).

nature dictates its reuse. The panel’s decision would eliminate the possibility of transformative use in software in all but completely non-functional uses. Software is inherently functional; unlike in other creative works, APIs in isolation can only serve one functional purpose because they are technical specifications with defined meanings. Thus, an API reimplementation—a functional reuse—must necessarily inherit functional characteristics of the original work, and the analysis of whether it is transformative therefore must examine the resulting new work *as a whole*.

## **II. Software APIs are inherently functional and are not copyrightable.**

As amici explained during the prior appeal, from a technical, industry history, and legal perspective, APIs are not copyrightable. *See* Computer Scientists Am. Br. I. The panel’s contrary copyrightability ruling is at odds with both the reality of software development and Ninth Circuit precedent. Because APIs are almost entirely functional, they are unprotectable as a “process, system, or method of operation.” *See* 17 U.S.C. § 102(b). And as explained in detail above, an API is a functional tool, and even if it has creative aspects, reimplementing an API implicates almost exclusively the functional ones.

The Ninth Circuit decisions in *Sega* and *Sony* recognize implicitly that the APIs, as functional tools, are not within the ambit of copyright. Both cases involved copying actual implementing code in order to discover—and, in *Sony*, to reimplement—APIs not unlike the ones at issue here. *Sega* at 1514-15; *Sony* at

605-06. Both focused on separating out the “functional concepts embodied in the code” for which copying “is lawful under section 102(b),” *Sega* at 1517; *see also Sony* at 603. Yet neither considered for a moment that the API itself, which was the ultimate goal of the challenged copying, could be protectable. The panel’s earlier decision on copyrightability cannot be reconciled with this precedent.

This error, if not corrected, will undermine decades of settled expectations and industry practice resulting in extraordinary innovation in the software industry. As amici explained in a prior brief, uncopyrightable interfaces were essential to the development of modern computers and the internet. *Computer Scientists Am. Br. II* at 4. Reimplementation isn’t just an accident—it’s fundamental to the very concept of APIs. *Id.* at 5. And the freedom to reimplement APIs has given the public enormous benefits, enabling the creation of new software and systems.

Since the birth of modern computing, progress and innovation in the software industry has been predicated on the open nature of APIs. Time and time again, API reimplementation has spurred innovation. *See id.* at 7. To take just one example, without the ability to freely reimplement an API, we would not have “PC” computers; the development of the entire market was based on the reimplementation of IBM’s API—to achieve the same functionality—by competing companies. Treating APIs as copyrightable threatens to shut down such future innovation. The full Circuit should grant rehearing to correct this error.

Dated: June 12, 2018

Respectfully Submitted,

/s/ Phillip R. Malone  
Phillip R. Malone  
Jeffrey T. Pearlman  
Juelsgaard Intellectual Property &  
Innovation Clinic  
Mills Legal Clinic at Stanford Law School  
559 Nathan Abbott Way  
Stanford, CA 94305  
Telephone: 650-725-6369  
Fax: 650-723-4426

*Attorneys for Amici Curiae*

## CERTIFICATE OF SERVICE

I hereby certify that on June 12, 2018, I electronically filed foregoing Brief of Computer Scientists as *Amici Curiae* in Support of Petition for Rehearing En Banc with the Clerk of the Court for the United States Court of Appeals for the Federal Circuit by using the appellate CM/ECF system.

I certify that all participants in the case are registered CM/ECF users and that service will be accomplished by the appellate CM/ECF system.

Dated: June 12, 2018

/s/ Phillip R. Malone  
Phillip R. Malone  
*Attorney for Amici Curiae*

## APPENDIX—LIST OF AMICI CURIAE

(In alphabetical order)

Amici are signing this brief on their own individual behalf and not on behalf of the companies or organizations with whom they are affiliated. Those affiliations are only for identification. This includes those amici indicated by an asterisk (\*), who are presently Google employees, consultants, and/or directors, and four amici, indicated by (†), who were retained as experts by Google or testified as fact witnesses (unpaid, in the case at least of Joshua Bloch) at one or both of the trials in this case. Those amici are signing this brief as individual computer scientists whose work in the field long preceded their affiliation with Google or this case. They are not signing this brief on behalf of Google or at Google's request.

1. Tom Ball.\* Tom Ball is a Staff Engineer at Google, working on Java-based developer tools. He was previously a Distinguished Engineer at Sun Microsystems, and a member of the JDK team that first released Java publicly. He wrote the first Java debugger (jdb), was a member of the AWT and Swing teams, and developed the Jackpot automated refactoring tool designed by James Gosling. His current project is J2ObjC (<http://j2objc.org>), an open source tool that converts Java source to Objective-C for use by iOS applications (which cannot run Java).
2. Brian Behlendorf. Brian Behlendorf is Executive Director of Hyperledger, an open source blockchain technology collaborative based at the Linux Foundation. He also serves as Chairman of the Board of the Electronic Frontier Foundation, and a member of the boards of the Mozilla Foundation and Benetech. He also co-founded the Apache Software Foundation, has worked as CTO for the World Economic Forum, advised and served the White House on open data and open source software issues, and co-founded a string of successful startups.
3. Jon Bentley. Jon Bentley's research interests include programming techniques, algorithm design, and the design of software tools and interfaces. He has written three books on programming and over a

hundred articles on a variety of topics, ranging from the theory of algorithms to software engineering. He received a B.S. from Stanford in 1974 and an M.S. and Ph.D. from the University of North Carolina in 1976, then taught Computer Science at Carnegie Mellon for six years. He joined Bell Labs Research in 1982, where he became a Distinguished Member of Technical Staff. He left Bell Labs in 2001 to join Avaya Labs research, from which he retired in 2013. He has been a visiting faculty member at West Point and Princeton, and has been a member of teams that have shipped software tools, telephone switches, telephones and web services. He holds over 40 US Patents. In March 2000 he received the Dr. Dobb's Excellence in Programming Award for advancing the craft of computer programming.

4. Matthew Bishop. Matthew Bishop received his Ph.D. in computer science from Purdue University, where he specialized in computer security, in 1984. He is on the faculty at the Department of Computer Science at the University of California at Davis. His main research area is the analysis of vulnerabilities in computer systems, including modeling, detecting, and analyzing them. Currently, he has research projects involving data sanitization, modeling election processes, and analyzing attacks. He is co-leading an education project aimed at improving the practice of programming using a “secure programming clinic” to help students improve the robustness and security of their programs. He has been active in the area of UNIX security since 1979, and has presented tutorials at SANS, USENIX, and other conferences. He also has done work on electronic voting, and was one of the two principle investigators of the California Top-to-Bottom Review, which performed a technical review of all electronic voting systems certified for use in the State of California. His textbook, *Computer Security: Art and Science* (Addison-Wesley, 2002), is used at many academic institutions throughout the world.
5. Joshua Bloch.<sup>†</sup> Joshua Bloch is an expert on API design, with over a quarter century of experience. He is a Professor of Computer Science at Carnegie Mellon University. Previously, he was Chief Java Architect at Google, a Distinguished Engineer at Sun Microsystems, and a Senior Systems Designer at Transarc

Corporation. He led the design and implementation of numerous Java APIs and language features, including the award-winning Java Collections Framework. He is the author of several books, including the bestselling, Jolt Award-winning *Effective Java* (Addison-Wesley, 2001, 2008, 2018), the de facto standard guide to Java best practices. He served on the National Academies CSTB Certifiably Dependable Software Committee. He holds a B.S. from Columbia and a Ph.D. in Computer Science from Carnegie Mellon University.

6. Dan Boneh. Dan Boneh is a Professor of Computer Science at Stanford University, where he heads the applied cryptography group. Dr. Boneh's research focuses on applications of cryptography to computer security. His work includes cryptosystems with novel properties, security for mobile devices, web security, and cryptanalysis. He is the author of over a hundred publications in the field and is a recipient of the 2013 Gödel prize, the Packard Award, the Alfred P. Sloan Award, the RSA award in mathematics and five best paper awards. In 2011 Dr. Boneh received the Ishii award for industry education innovation. Dr. Boneh's wife is a current Google employee.
7. Gilad Bracha. Gilad Bracha is the creator of the Newspeak programming language and a well known researcher in the area of object-oriented programming languages. He was awarded the senior Dahl-Nygaard prize in 2017. Previously, he has worked at Google, as a VP at SAP Labs in Palo Alto, a Distinguished Engineer at Cadence, and a Computational Theologist and Distinguished Engineer at Sun. He has authored or co-authored several books including the Java Language and Virtual Machine Specifications, and the Dart Programming Language. Prior to joining Sun, he worked on Strongtalk, the Animorphic Smalltalk System. He received his B.Sc in Mathematics and Computer Science from Ben Gurion University in Israel and a Ph.D. in Computer Science from the University of Utah.
8. Eric Brewer.\* Eric Brewer pioneered early "cloud" computing starting in the 1990s with research on large-scale services implemented on clusters of commodity servers, for which he was

elected to the National Academy of Engineering. In 1996, Brewer co-founded Inktomi Corporation, an early search engine that also influenced the modern Internet architecture. He formulated the CAP theorem, one the tenets of modern distributed systems. In 2000, working with President Clinton, he led the development of usa.gov, the primary federal portal. He is a tenured professor in the Computer Science department at UC Berkeley, but is currently on leave at Google as VP, Infrastructure. Brewer received a BS in EECS from UC Berkeley, and an M.S. and Ph.D. from MIT.

9. Frederick Brooks. Brooks is the Kenan Professor of Computer Science (Emeritus) at University of Northern Carolina at Chapel Hill. As Corporate Project Manager for IBM's System/360 (mainframe) computer family hardware and the Operating System/360 software, he in 1964 switched the standard computer byte size from 6 to 8 bits. He was an architect of the Stretch and Harvest supercomputers. He founded UNC's Computer Science Department. He's researched computer architecture, software engineering, the design process, and graphics virtual environments. He wrote *The Mythical Man-Month*, *The Design of Design*, and with G.A. Blaauw, *Computer Architecture*. Honors include the National Medal of Technology, the ACM Turing award, the National Academies of Engineering and Science, and British and Dutch academies.
10. Rick Cattell.† R. G. G. "Rick" Cattell is an independent consultant in database systems. He previously worked as a Distinguished Engineer at Sun Microsystems. Dr. Cattell served for 20 years at Sun Microsystems in management and senior technical roles, and for 10 years in research at Xerox PARC and Carnegie Mellon University. He is best known for his contributions in database systems and middleware, including database scalability, Enterprise Java, object/relational mapping, object-oriented databases, and database interfaces. At Sun he instigated Enterprise Java, JDBC, Java DB, and Java Blend, and contributed to many Java APIs and products. He previously developed Xerox PARC's Cedar DBMS, Sun's Simplify database GUI, and SunSoft's CORBA-database integration. He is a co-founder of SQL Access (predecessor to ODBC), founder and chair of the Object Data Management Group

(ODMG), author of the world's first monograph on object/relational and object databases, recipient of the ACM Outstanding Ph.D. Dissertation Award, and an ACM Fellow.

11. Vinton G. Cerf.\* Vinton G. "Vint" Cerf is vice president and Chief Internet Evangelist for Google, where he contributes to global policy development and the continued spread of the Internet. Widely known as one of the "Fathers of the Internet," Cerf is the co-designer of the TCP/IP protocols and the architecture of the Internet. He has served in executive positions at MCI, the Corporation for National Research Initiatives, the Defense Advanced Research Projects Agency, and on the faculty of Stanford University. Cerf served as chairman of the board of the Internet Corporation for Assigned Names and Numbers (ICANN) from 2000-2007. Cerf is a Fellow of the IEEE, ACM, and AAAS, the American Academy of Arts and Sciences, the International Engineering Consortium, the Computer History Museum, and is a member of the National Academy of Engineering. He is a former President of the ACM and Founding President of the Internet Society. President Obama appointed him to the National Science Board in 2012. Cerf is a recipient of numerous awards and commendations in connection with his work on the Internet, including the US Presidential Medal of Freedom, US National Medal of Technology, the Queen Elizabeth Prize for Engineering, the ACM Turing Award, Officer of the Legion d'Honneur and 29 honorary degrees. In December 1994, People magazine identified Cerf as one of that year's "25 Most Intriguing People." Cerf holds a B.S. from Stanford, and an M.S. and Ph.D. from UCLA.
12. William Cook.\* William Cook is an Associate Professor in the Department of Computer Sciences at the University of Texas at Austin. His research is focused on object-oriented programming, programming languages, modeling languages, and the interface between programming languages and databases. Prior to joining UT in 2003, Dr. Cook was Chief Technology Officer and co-founder of Allegis Corporation. He was chief architect for several award-winning products, including the eBusiness Suite at Allegis, the Writer's Solution for Prentice Hall, and the AppleScript language at Apple Computer. At HP Labs his research focused on the

foundations of object-oriented languages, including formal models of mixins, inheritance, and typed models of object-oriented languages. He completed his Ph.D. in Computer Science at Brown University in 1989. He received the Dahl-Nygaard Senior Prize in 2014 for his contributions to the theory and practice of object-oriented programming.

13. Mark Davis.\* Dr. Mark Davis has been the Chief Internationalization Architect at Google since 2006, focusing on effective and secure use of Unicode, software internationalization libraries, and related areas. Dr. Davis is also the co-founder and has been president of the Unicode Consortium since its inception in 1991, and is a key technical contributor to the Unicode specifications. In 2003, he founded the Unicode Common Locale Data Repository (CLDR) project, the standard repository for locale data worldwide. He is co-author of BCP 47 (“Tags for Identifying Languages”), used to identify human languages in all XML and HTML documents, and in all modern programming libraries. Mark provided the original architecture of ICU, the premier Unicode software internationalization library, and the Java internationalization libraries. At IBM, he was Chief Software Globalization Architect. At Taligent, he was manager and architect for the international frameworks. At Apple, he co-authored the first Macintosh system to support Japanese (KanjiTalk), and authored the first Macintosh Arabic and Hebrew systems. Mark holds a Ph.D. from Stanford University and a B.A. from the University of California, Irvine.
14. Jeffrey Dean.\* Jeffrey Dean joined Google in 1999 and is currently one of two Senior Fellows in the company, where he leads the Google Brain team, Google’s artificial intelligence research team. He has co-designed/implemented five generations of Google’s crawling, indexing, and query serving systems, and co-designed/implemented major pieces of Google’s initial advertising and AdSense for Content systems. He is also a co-designer and co-implementor of Google’s distributed computing infrastructure, including the MapReduce, BigTable and Spanner systems, protocol buffers, LevelDB, systems infrastructure for statistical machine translation, the TensorFlow open-source machine learning system, and a variety of internal and

external libraries and developer tools. Prior to joining Google, Jeff did computer systems research at Digital Equipment Corporation's Western Research Lab. Jeff has also worked for both the Centers for Disease Control and the World Health Organization, designing computer software for epidemiology and for statistical analysis of the HIV/AIDS pandemic. He is a Fellow of the ACM and the AAAS, a member of the U.S. National Academy of Engineering, and a recipient of the Mark Weiser Award and the ACM-Infosys Foundation Award in the Computing Sciences. Jeff holds a B.S., summa cum laude, in computer science and economics from the University of Minnesota, and an M.S. and Ph.D. in computer science from the University of Washington.

15. L Peter Deutsch. Dr. L Peter Deutsch received a Ph.D. in Computer Science from U.C. Berkeley in 1973. Subsequently at Xerox PARC, he helped develop the Interlisp-D, Cedar Mesa, and Smalltalk-80 programming systems. Deutsch's work on Smalltalk implementation, among other innovations, was an important contributor to the just-in-time compilation technology now used widely to dramatically improve the performance of Java and JavaScript implementations. He is also the author of a number of RFCs and of *The Eight Fallacies of Distributed Computing*, and originated the Deutsch limit adage about visual programming languages. From 1986 to 1991, as Chief Scientist at ParcPlace Systems, he developed cross-platform JIT technology. From 1986 to 2003, dba Aladdin Enterprises, he was the creator of Ghostscript, an Open Source implementation of the PostScript language. In 1993, he was a co-recipient of the ACM Software System Award, and was also named a Distinguished Alumnus of the U.C. Berkeley Computer Science program; he was named an ACM Fellow in 1994. In 1994, he founded Artifex Software to license Ghostscript commercially while continuing its development and its release as Open Source; Artifex today is a multi-million-dollar business. In 1999-2000, he served on the board of the Open Source Initiative. He is a co-inventor on two patents.
16. David L. Dill. David Dill is The Donald E. Knuth Professor, Emeritus, in the School of Engineering at Stanford University. Prof. Dill's Ph.D. thesis, "Trace Theory for Automatic Hierarchical

Verification of Speed Independent Circuits” was named as a Distinguished Dissertation by the Association for Computing Machinery (ACM), and published as such by M.I.T. Press in 1988. He was named a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) in 2001 for his contributions to verification of circuits and systems, and a Fellow of the ACM in 2005 for contributions to system verification and for leadership in the development of verifiable voting systems. In 2008, he received the first “Computer-Aided Verification” award for fundamental contributions to the theory of real-time systems verification. In 2013, he was elected to the National Academy of Engineering and the American Academy of Arts and Sciences. In 2016, he received the Alonzo Church Award for Outstanding Contributions to Logic and Computation.

17. Lester Earnest. Lester Earnest is a widely-recognized computer scientist, best known for his deep involvement with the Advanced Research Project Agency Network (ARPAnet) startup committee, which led to his invention of the Finger social networking protocol. He served as a US Navy Aviation Electronics Officer and Digital Computer Project Officer at the Naval Air Development Center, and later joined MIT to help design the Semi-Automatic Ground Environment air defense system. Later, he innovated numerous early features in the nascent field of word processing, including the first spell-checker, search engine, self-driving vehicle, robotic hand-eye assembler that took verbal instructions, online restaurant reviews, online news service, and a number of other successful innovations.
18. Dawson Engler. Dawson Engler is an Associate Professor at Stanford. He received his Ph.D. from MIT for his work on the exokernel operating system and his undergraduate degree from University of Arizona. His research focuses on devising automatic methods to find as many interesting bugs in real code as possible, including static analysis, implementation level model checking, and symbolic execution. His research group has won numerous “Best Paper” awards. Its early static tools have found millions of errors in mature open source and commercial systems and have formed the basis of a successful company, Coverity. His group’s more recent

tool, KLEE, is a symbolic execution system widely used in the research community. He won the 2006 Weiser award and the 2008 ACM Grace M Hopper award.

19. Martin Fowler. Martin Fowler is an author and educator on software development. He is currently chief scientist at ThoughtWorks, a global system delivery and consulting firm. Mr. Fowler concentrates on the design of enterprise software: what makes a good design and what practices are needed to enhance it. He is the author of seven books on software development, which have over a million copies in print in over a dozen languages. He is the editor of a book series with Addison-Wesley on software design. His website, <http://martinfowler.com>, is a wide-ranging resource of software development techniques attracting around 150,000 visitors per month.
20. Neal Gafter. Neal Gafter is a Principal Engineer at Microsoft, where he is a technical lead for the Roslyn Project (Microsoft's implementation for the C# and Visual Basic programming languages). Previously he was a software engineer and Java Evangelist at Google, where he designed and implemented the distributed storage architecture for Google Calendar, and a Senior Staff Engineer at Sun Microsystems, where he led the development of the Java compiler and implemented the Java language features in releases 1.4 through 5.0. Neal was a member of the C++ Standards Committee and led the development of C and C++ compilers at Sun Microsystems, Microtec Research, and Texas Instruments. He holds a B.S. in computer engineering from Case Western Reserve University and a Ph.D. in computer science from the University of Rochester.
21. Robert Harper. Robert Harper is a professor in the computer science department at Carnegie Mellon University. He holds a Ph.D. in computer science from Cornell University. His main research interest is in the application of type theory to the design and implementation of programming languages and to the mechanization of their meta-theory. Harper made major contributions to the design of the Standard ML programming language and the LF logical framework. Harper is a recipient of the

Allen Newell Medal for Research Excellence and the Herbert A. Simon Award for Teaching Excellence, and is an Association for Computing Machinery Fellow.

22. John Hennessy.\* John Hennessy is a Professor of Electrical Engineering and Computer Science and Director of the Knight-Hennessy Scholars Program at Stanford University. Professor Hennessy served as President of Stanford University for sixteen years. He serves on the boards of Google, Cisco Systems, and the Gordon and Betty Moore Foundation. Professor Hennessy is an IEEE Fellow, a member of the National Academy of Sciences and the National Academy of Engineering, and a Fellow of the American Academy of Arts and Sciences and the Association for Computing Machinery. He shared the IEEE von Neumann Medal, the NEC C&C Prize, and the ACM Turing Award with David Patterson. He is the co-author of two internationally used undergraduate and graduate textbooks on computer architecture design.
23. Alan Kay. Alan Kay is one of the pioneers of object-oriented programming, personal computing, and graphical user interfaces. For this work, Dr. Kay has received the Draper Prize from the National Academy of Engineering, the ACM Turing Award, and the Kyoto Prize from the Inamori Foundation. Alan has been elected a fellow of the American Academy of Arts and Sciences, the National Academy of Engineering, the Royal Society of Arts, the AAAS, and the Computer History Museum. Alan has held fellow positions at HP, Disney, Apple, and Xerox, and has served as the chief scientist at Atari. While at Xerox PARC, he was one of the key members there to develop prototypes of networked workstations using the programming language Smalltalk. He is an adjunct professor of computer science at UCLA and an advisor to One Laptop per Child. At Viewpoints Research, Alan also continues his work with “powerful ideas education” for the world’s children, as well as the development of advanced personal computers and networking systems.
24. Brian Kernighan.\* Brian Kernighan is a professor in the Computer Science Department of Princeton University. He worked at Bell Labs alongside Unix creators Ken Thompson and Dennis Ritchie

and contributed to the development of Unix. He co-authored a number of Unix programs, including widely used document preparation tools. He is also the author or co-author of 11 books on computing, including the first book on the C programming language with Dennis Ritchie; these books have been translated into more than two dozen languages. He is also a co-creator of the AWK and AMPL programming languages. In collaboration with Shen Lin he devised well-known heuristics for two fundamental NP-complete optimization problems: graph partitioning and the traveling salesman problem. Kernighan received a Bachelor's degree in engineering physics from the University of Toronto, and his Ph.D. in electrical engineering from Princeton University. He is a member of the National Academy of Engineering.

25. David Klausner. David Klausner has over 50 years of software/hardware development and consulting experience in the computer and software industry. He has written software for commercial products as an engineer, developer, supervisor, project manager, department manager, middle manager and company executive. He has worked in forensic investigation and in reverse engineering. He has been employed in various capacities for various companies, such as Microsoft, AT&T, Cisco, IBM, Hewlett Packard, and Intel Corporation. He holds a Bachelors of Arts degree in Mathematics, and a Master of Science degree in Electrical Engineering. He has taught programming, public speaking, has guest lectured at Stanford University, and been an invited speaker by IBM, AT&T, and others. His technical opinions have been confirmed in several cases by the United States Court of Appeals for the Federal Circuit.
26. Kin Lane. Kin is a computer scientist and API Evangelist working to understand the technology, business and politics of APIs and help share this insight with the world. He is the author of the book, Business of APIs, and is behind the popular API Evangelist blog. He has over 20 years of experience as a programmer, database administrator, architect, product developer, manager, and executive in the API space.

27. Ed Lazowska. Ed Lazowska holds the Bill & Melinda Gates Chair in the Paul G. School of Computer Science & Engineering at the University of Washington. His research concerns the design, implementation, and analysis of high performance computing and communication systems, and, more recently, the techniques and technologies of data-intensive discovery. He co-chaired (with Marc Benioff) the President's Information Technology Advisory Committee from 2003-05, and (with David E. Shaw) the Working Group of the President's Council of Advisors on Science and Technology to review the Federal Networking and Information Technology Research and Development Program in 2010. He is a Member of the National Academy of Engineering and a Fellow of the American Academy of Arts and Sciences.
28. Doug Lea. Doug Lea is a Professor of Computer Science at the State University of New York at Oswego. He is an author of books, articles, reports, and standardization efforts on object oriented software development including those on specification, design and implementation techniques, distributed, concurrent, and parallel object systems, and software reusability; he has served as chair, organizer, or program committee member for many conferences and workshops in these areas. He is the primary author of several widely used software packages and components.
29. Bob Lee.<sup>†</sup> Bob Lee is CEO of Present Company, makers of Present, a new location-based social network. Prior to that, as Square's CTO, Bob built Square's core products, scaled the team from 12 to 1200 people, and created Square Cash. Before Square, Bob worked at Google where he created Guice and was the core library lead for Android.
30. Paul Menchini. Paul Menchini is the Chief Information Security Officer at the North Carolina School of Science and Mathematics. Previously, he held technical positions at HP, Intel, GE Microelectronics, CLSI and OrCAD. As a member of the "Woods Hole Summer Study on Hardware Description Languages," he contributed to the specifications for VHDL; subsequently, he edited two revisions of IEEE Std 1076 VHDL and developed the first commercially successful VHDL compiler. As part of the compiler

project, he developed an API for a VHDL intermediate form, which was subsequently standardized by the IEEE. He holds a Masters in Computer Engineering from Stanford University and is the recipient of numerous technical awards, including charter membership in the “IEEE Golden Core.”

31. James H. Morris. Dr. James H. Morris is a Professor of Computer Science at Carnegie Mellon University, where he served as Dean of the Silicon Valley Campus, Dean of the School of Computer Science, Head of the Computer Science Department, and Director of the Information Technology Center, a joint project with IBM that developed a prototype university computing system. He founded Carnegie Mellon’s Human Computer Interaction Institute, Robot Hall of Fame, and Silicon Valley Campus. He was an Associate Professor at UC Berkeley, where he developed two fundamental principles of programming languages: inter-module protection and lazy evaluation. He was co-discoverer of the Knuth-Morris-Pratt string-searching algorithm. He was Principal Scientist and Research Fellow at Xerox PARC, where he was part of the team that developed the Alto, a precursor to today’s personal computers. He is a founder of MAYA Design Group and an ACM Fellow. He holds a B.S. from CMU and an M.S. and Ph.D. from MIT.
32. Peter Norvig.\* Peter Norvig is a Director of Research at Google; previously he directed Google’s core search algorithms group. He is co-author of *Artificial Intelligence: A Modern Approach*, the leading textbook in the field, and co-teacher of an Artificial Intelligence class that signed up 160,000 students, helping to kick off the current round of massive open online classes (MOOCs). He is a fellow of the AAI, ACM, California Academy of Science and American Academy of Arts and Sciences.
33. Martin Odersky. Martin is a professor at EPFL in Lausanne, Switzerland. He is best known as the creator and principal designer of the Scala programming language. Prior to that, he made several contributions to the development of Java. He created the Pizza and GJ languages, designed the original version of generics for Java, and wrote the javac reference compiler for Java. He is a fellow of the ACM.

34. Tim Paterson. Tim began his career designing one of the first 16-bit microcomputer systems at Seattle Computer Products. He then wrote an operating system for that computer, which was later sold to Microsoft and became widely used as MS-DOS. He went on to found his own company, Falcon Technology, whose primary products were hard disk systems for personal computers. He moved on to Microsoft where he was a software engineer for many years, working on such products as QuickBASIC, Visual Basic, VBScript, and Visual J++ (Java). After his retirement in the late '90s he has continued developing software and microcontroller-based hardware projects as a hobby and part-time small business. He has been granted three U.S. patents on software methods.
35. David Patterson.\* David Patterson joined UC Berkeley in 1977. He has been Director of the Par Lab, Chair of UC Berkeley's CS Division, Chair of the Computing Research Association, and President of the Association for Computing Machinery. His most successful projects have been Reduced Instruction Set Computers (RISC), Redundant Arrays of Inexpensive Disks (RAID), and Network of Workstations. All helped lead to multibillion-dollar industries. This research led to many papers, six books, and about 35 honors, including election to the National Academy of Engineering, the National Academy of Sciences, the Silicon Valley Engineering Hall of Fame, and Fellow of the Computer History Museum. He shared the IEEE von Neumann Medal, the NEC C&C Prize, and the ACM Turing Award (the highest prize in computing) with John Hennessy, former President of Stanford University and co-author of two of his books.
36. Alex Payne. Alex Payne consults, advises, and invests in early-stage technology startups. As Platform Lead at Twitter he managed one of the web's most popular APIs. He was subsequently co-founder and Chief Technology Officer of online banking service Simple, acquired by BBVA in 2014. Alex organizes an annual conference showcasing advances in programming languages and has co-authored a book on the Scala programming language (O'Reilly, 2009). He is a regular speaker at technology and business conferences worldwide and has lectured on API design at Stanford.

37. Tim Peierls. Since receiving a BS in Computer Science from Yale in 1983 and an MS in CS from Cornell in 1986, Tim has continuously worked in the software industry, first at Bell Labs (airline crew scheduling), then co-founding the Lightstone Group in 1990 (aircraft scheduling, delivery vehicle routing and scheduling, sold to Descartes Systems Group in 1998) and Seat Yourself in 2002 (online ticketing for school performing arts groups). For the last fifteen years, almost all of his programming work has been in Java. He has served on the Expert Groups of several Java Specification Requests (166, 201, 330, 334) and on the SE/EE Executive Committee of the Java Community Process; he co-authored a book, *Java Concurrency in Practice*; and he contributes code, support, and advice to various open source projects, including Restlet, Hazelcast, and JClouds.
38. Simon Phipps.<sup>†</sup> Simon is a director and president of the Open Source Initiative, the global steward of the Open Source Definition. OSI serves to advocate for, educate about, and build bridges within the open source community. His career has included early engagement in establishing Java, XML and weblogs as computer industry technologies as well as contributions to open standards in a variety of fields. As chief open source officer at Sun Microsystems he supervised the open source relicensing of Solaris Unix, Java and many other software systems. He is currently founder and CEO of Meshed Insights Ltd, a UK firm offering management services related to open source and digital rights.
39. Ronald L. Rivest. Ronald L. Rivest is an MIT Institute Professor in the Electrical Engineering and Computer Science Department. He is well-known as a co-inventor of the RSA public-key cryptosystem, for which he received the ACM Turing Award. He is a co-author of the widely-used textbook *Introduction to Algorithms*. His current research interest is voting systems and election integrity.
40. Curtis Schroeder. Curtis is a Hardware-in-the-Loop Simulation Engineer at Draper. He served as the Drafting Group Editor for the Simulation Interoperability Standards Organization (SISO) Common Image Generator Interface (CIGI) 4.0 international standard. The success of SISO international standards depends upon implementation of said copyrighted standards by numerous

simulation vendors and end-users, including NATO. Previously, Curtis has worked for Antycip Simulation in the UK and the Lockheed Martin Aeronautics Company, where he utilized a number of open standards in projects he was involved in. He earned B.S. & M.S. Computer Science degrees at the Missouri University of Science & Technology.

41. Robert Sedgewick. Robert Sedgewick is the founding chair and the William O. Baker Professor in the Department of Computer Science at Princeton and served for many years as a member of the board of directors of Adobe Systems. He has over 50 years of experience working with software systems. He has held visiting research positions at Xerox PARC, Palo Alto, CA; Institute for Defense Analyses, Princeton, NJ; and INRIA, Rocquencourt, France. He regularly serves on journal editorial boards and organizing program committees of conferences and workshops on data structures and the analysis of algorithms held throughout the world. Professor Sedgewick's research interests include analytic combinatorics, algorithm design, the scientific analysis of algorithms, curriculum development, and innovations in the dissemination of knowledge. He has published widely in these areas and is the author of twenty books, including a series of books on algorithms that have been bestsellers for four decades and have sold nearly one million copies. He has also published extensive online content (including studio-produced video lectures) on analysis of algorithms and analytic combinatorics and (with Kevin Wayne) algorithms and computer science. Their MOOC on algorithms has been named one of the "top 10 MOOCs of all time."
42. Mary Shaw. Mary Shaw is the Alan J. Perlis University Professor of Computer Science in the Institute for Software Research at Carnegie Mellon University. Her research focuses on software engineering and software design, particularly software architecture and design of systems used by real people. She has made fundamental and significant contributions to an engineering discipline for software through developing data abstraction with verification, establishing software architecture as a major branch of software engineering, designing influential and innovative curricula in software engineering and computer science supported by two

influential textbooks, and helping to found the Software Engineering Institute at Carnegie Mellon. She has received the United States' National Medal of Technology and Innovation, the George R. Stibitz Computer & Communications Pioneer Award, the ACM SIGSOFT Outstanding Research Award, the IEEE Computer Society TCSE's Distinguished Educator and Distinguished Women in Software Engineering Awards, and CSEE&T's Nancy Mead Award for Excellence in Software Engineering Education. She is an elected Life Fellow of the ACM and the IEEE and an elected Fellow of the AAAS. She holds a BA *cum laude* from Rice and a Ph.D. from Carnegie Mellon.

43. Dave Snigier. Dave Snigier is the technical lead for Massachusetts Advanced Secure Technologies (MAST) and an enterprise architect at the University of Massachusetts Office of the President, designing systems to help keep public higher education cost-effective and secure. He has led several successful projects including a UMass paperless initiative and an award-winning managed security offering for institutions across the state of Massachusetts.
44. Alfred Z. Spector.\* Alfred Spector is Chief Technology Officer and Head of Engineering at Two Sigma, a firm dedicated to using information to optimize diverse economic challenges. Prior to joining Two Sigma, Dr. Spector spent nearly eight years as Vice President of Research and Special Initiatives, at Google, where his teams delivered a range of successful technologies including machine learning, speech recognition, and translation. Prior to Google, Dr. Spector held various senior-level positions at IBM, including Vice President of Strategy and Technology (or CTO) for IBM Software and Vice President of Services and Software research across the company. He previously founded and served as CEO of Transarc Corporation, a pioneer in distributed transaction processing and wide-area file systems, and he was a professor of computer science at Carnegie Mellon University. Dr. Spector received a bachelor's degree in Applied Mathematics from Harvard University and a Ph.D. in computer science from Stanford University. He is a Fellow of both the Association for Computing Machinery and the IEEE. He is an active member of the National Academy of Engineering and the American Academy of Arts and Sciences, where he serves on the

Council. Dr. Spector won the IEEE Kanai Award for Distributed Computing in 2001 and the ACM Software Systems Award for his work on the Andrew File System (AFS) in 2016.

45. Bjarne Stroustrup. Bjarne Stroustrup is the designer and original implementer of C++ as well as the author of *The C++ Programming Language (Fourth Edition)*, *A Tour of C++, Programming: Principles and Practice using C++ (Second Edition)*, and many popular and academic publications. Dr. Stroustrup is a Managing Director in the technology division of Morgan Stanley in New York City as well as a visiting professor at Columbia University. He is a member of the US National Academy of Engineering, and an IEEE, ACM, and CHM fellow. He received the 2018 Charles Stark Draper Prize, the IEEE Computer Society's 2018 Computer Pioneer Award, and the 2017 IET Faraday Medal. His research interests include distributed systems, design, programming techniques, software development tools, and programming languages. He is actively involved in the ISO standardization of C++. He holds a masters in Mathematics from Aarhus University, where he is an honorary professor, and a PhD in Computer Science from Cambridge University, where he is an honorary fellow of Churchill College.
46. Ivan E. Sutherland. Ivan E. Sutherland received his B.S. degree from the Carnegie Institute of Technology, his M.S. degree from the California Institute of Technology, and his Ph.D. degree from the Massachusetts Institute of Technology, all in electrical engineering. He holds honorary degrees from Harvard University, the University of North Carolina, and the University of Utah. He joined Sun in 1990 as a Sun Fellow, Sun's highest technical rank. He joined Portland State University in 2009 to found the Asynchronous Research Center. He leads a small group working on self-timed VLSI systems; his group develops self-timed circuit methodologies and design techniques for fast CMOS circuits and applies them to new hardware architectures. His book, *Logical Effort* (1999) with joint authors Sproull and Harris, describes the mathematics of designing fast circuits. His 1963 MIT Ph.D., *Sketchpad*, is widely known, and he has been called the "father of computer graphics." He is author of more than 70 patents, as well as numerous publications and lectures. Dr. Sutherland holds the 1988 ACM Turing Award,

the 2012 Kyoto Prize and the IEEE Von Neumann Award. He is a Fellow of the ACM and a member of both the National Academy of Engineering and the National Academy of Sciences.

47. Brad Templeton. Brad Templeton, active in the computer network community since 1979, was founder and publisher at ClariNet Communications Corp., the electronic newspaper that was perhaps the earliest dot-com company. He participated in the building and growth of USENET from its earliest days, and in 1987 founded and edited rec.humor.funny, for many years the world's most widely read electronic publication. He was the first employee of Personal Software/Visicorp, the first major microcomputer applications software company. He later founded Looking Glass Software and over the years was author of a dozen packaged microcomputer software products, including VisiPlot for the IBM-PC, various games, popular tools and utilities for Commodore computers, special Pascal and Basic programming environments designed for education (ALICE), an add-in spreadsheet compiler for Lotus 1-2-3 (3-2-1 Blastoff), and various network related software tools. He currently is track chair for computing and networks at Singularity University, a consultant and speaker on self-driving cars, and is on the board of the Electronic Frontier Foundation and the Foresight Nanotech Institute. He is Chairman Emeritus of the Electronic Frontier Foundation.
48. Ken Thompson.\* Ken Thompson spent much of his career at Bell Laboratories where he co-designed and implemented the original Unix operating system, invented the B programming language that was a precursor to the C programming language, invented the Bon programming language, co-developed the Plan 9 operating systems, developed the CTSS version of the editor QED, developed ed, which is the standard text editor on Unix, and the definition of the UTF-8 encoding, which is used for more than half of all Web pages. Thompson also co-developed the software and hardware for Belle, which was the first computer built for the sole purpose of chess playing, and it officially became the first master-level machine in 1983. He is currently a Google Advisor and was formerly a Distinguished Engineer at Google, where he invented new programming languages (including the Go programming language

as a co-inventor), among other projects. Thompson is a recipient of numerous awards and commendations in connection with his work on Unix, including the IEEE Emanuel R. Piore Award (1982), the Turing Award (1983), the IEEE Richard W. Hamming Medal (1990), the National Medal of Technology (1999), and the Japan Prize (2011). He is a member of the National Academy of Sciences and the National Academy of Engineering. Thompson holds a B.S. and an M.S., both in Electrical Engineering and Computer Science, from the University of California, Berkeley. He has been awarded two honorary Ph.D degrees.

49. Andrew Tridgell. Dr. Andrew Tridgell is a computer scientist and free software developer in Canberra, Australia. Best known for his contributions to the development of the award winning Samba suite of networking software that enables interoperability with Microsoft networking services, he has been actively developing in the area of interoperability for more than 20 years.
50. Jeffrey Ullman. Jeffrey Ullman is the Stanford W. Ascherman Professor of Engineering (Emeritus) in the Department of Computer Science at Stanford and CEO of Gradiance Corp. He received a B.S. degree from Columbia University in 1963 and a Ph.D. from Princeton in 1966. Prior to his appointment at Stanford in 1979, he was a member of the technical staff of Bell Laboratories from 1966-1969, and on the faculty of Princeton University between 1969-1979. From 1990-1994, he was chair of the Stanford Computer Science Department. Ullman was elected to the National Academy of Engineering in 1989, the American Academy of Arts and Sciences in 2012, and has held Guggenheim and Einstein Fellowships. He has received the Sigmod Contributions Award (1996), the ACM Karl V. Karlstrom Outstanding Educator Award (1998), the Knuth Prize (2000), the Sigmod E. F. Codd Innovations award (2006), the IEEE von Neumann medal (2010), and the NEC C&C Prize (2017). He is the author of 16 books, including books on database systems, compilers, automata theory, and algorithms.
51. Andries van Dam. Andries van Dam is a Professor of Computer Science at Brown University, and has served on Brown's Computer Science faculty since 1965. He was also Brown's first Vice President

of Research from 2002 to 2006. He is the author of the widely used reference books *Computer Graphics: Principles and Practice* and *Object-Oriented Programming Java: A Graphical Approach*. In 1967, Andries co-founded ACM SIGGRAPH, the precursor to SIGGRAPH. Andries is an IEEE Fellow, an ACM Fellow, and is a member of the National Academy of Engineering and the American Academy of Arts and Sciences. Andries has won multiple awards, including the Information Display's Special Recognition Award (1974), the IEEE Centennial Medal (1984), the National Computer Graphics Association's Academic Award (1990), the ACM SIGGRAPH Steven A. Coons Award (1991), the L. Herbert Ballou University Professor Chair (1992), the ACM Karl V. Karlstrom Outstanding Educator Award (1994), the Thomas J. Watson, Jr. University Professor of Technology and Education Chair (1995), the IEEE James H. Mulligan, Jr. Education Medal (1999), and the ACM SIGCSE Award for Outstanding Contributions to Computer Science Education (2000). Andries received a B.S. with honors in Engineering Science from Swarthmore College, a M.S. and Ph.D. from the University of Pennsylvania, and holds honorary Ph.D. degrees from Darmstadt Technical University, University of Waterloo, ETH Zurich, and Swarthmore College.

52. Guido van Rossum. Guido van Rossum created the open-source programming language Python, and is its lead developer and thought leader. Python is widely used in industry, and is the most popular introductory teaching language at top U.S. universities. Guido developed the Python language while at CWI in Amsterdam. After moving to the United States, he worked as a guest researcher at NIST, at CNRI, and at several start-up companies. He became a Senior Staff Engineer at Google, and is currently a principal engineer at Dropbox. Guido is an ACM Distinguished Engineer, a Fellow of the Computer History Museum, and a recipient of several awards including the USENIX STUG Award, the NLUUG Award, the Free Software Foundation Award, and the Dr. Dobb's Journal 1999 Excellence in Programming Award. In 2013, Python was awarded the Dutch National ICT COMMIT Award. Guido holds an M.S. in Mathematics and Computer Science from the University of Amsterdam.

53. John Villasenor. John Villasenor is on the faculty at UCLA, where he is a professor of electrical engineering, public policy, and management, as well as a visiting professor of law. He is also a nonresident senior fellow at the Brookings Institution and a visiting fellow at the Hoover Institution. Professor Villasenor's research considers communications and information technologies and their broader ramifications, and has addressed topics including cybersecurity, autonomous vehicles, and digital media policy. Professor Villasenor is a member of the Council on Foreign Relations and a former vice chair of the World Economic Forum's Global Agenda Council on the Intellectual Property System. He holds an M.S. and Ph.D. in electrical engineering from Stanford University, and a B.S. in electrical engineering from the University of Virginia. Professor Villasenor has previously served as, though is not currently serving as, a consultant to Google in relation to the *Oracle v. Google* matter.
54. Jan Vitek. Jan Vitek is a Professor of Computer Science at Northeastern University. He is the past Chair of the ACM Special Interest Group on Programming Languages (SIGPLAN), the vice chair of AITO and of the IFIP WG 2.4, and is Chief Scientist at Fiji Systems. He holds a Ph.D. from the University of Geneva and an MSc from the University of Victoria. He works on various aspects of programming languages including virtual machines, compilers, software engineering, real-time and embedded computing, concurrency and information security. Professor Vitek led the Ovm project which resulted in the first successful flight test of real-time Java virtual machine. With Noble and Potter, Vitek proposed the notion of ownership for alias control, which became known as ownership types. He chaired PLDI, ISMM and LCTES and was program chair of ESOP, ECOOP, VEE, Coordination, and TOOLS.
55. Philip Wadler. Philip Wadler is a Professor of Theoretical Computer Science at the University of Edinburgh and Senior Research Fellow at IOHK. He is an ACM Fellow and a Fellow of the Royal Society of Edinburgh, past chair of ACM SIGPLAN, past holder of a Royal Society-Wolfson Research Merit Fellowship, winner of the SIGPLAN Distinguished Service Award, and a winner of the POPL Most Influential Paper Award. Previously, he worked or studied at

Stanford, Xerox Parc, CMU, Oxford, Chalmers, Glasgow, Bell Labs, and Avaya Labs, and visited as a guest professor in Copenhagen, Sydney, and Paris. He has an h-index of 66 with more than 22,000 citations to his work, according to Google Scholar. He contributed to the designs of Haskell, Java, and XQuery, and is a co-author of *Introduction to Functional Programming* (Prentice Hall, 1988), *XQuery from the Experts* (Addison Wesley, 2004) and *Generics and Collections in Java* (O'Reilly, 2006). He has delivered invited talks in locations ranging from Aizu to Zurich.

56. James H. Waldo. James “Jim” Waldo is the Gordon McKay Professor of the Practice of Computer Science in the School of Engineering and Applied Sciences at Harvard, where he is also the Chief Technology Officer. Jim is also a professor of technology policy at the Harvard Kennedy School. Previously, Jim designed clouds at VMware, and was a Distinguished Engineer at Sun Microsystems, where he investigated next-generation large-scale distributed systems. He was the lead architect for Jini, a distributed programming system based on Java. Before joining Sun, Jim spent eight years at Apollo Computer and Hewlett Packard, working in the areas of distributed object systems, user interfaces, class libraries, text and internationalization. While at HP, he led the design and development of the first Object Request Broker, and was instrumental in getting that technology incorporated into the first OMG CORBA specification. Jim edited the book *The Evolution of C++: Language Design in the Marketplace of Ideas* (MIT Press), co-edited *Engaging Privacy and Information Technology in a Digital Age* (National Academies Press), and was one of the authors of *The Jini Specification* (Addison Wesley). More recently, he authored *Java: The Good Parts*. He is currently a member of the editorial boards of Queue magazine and Communications of the ACM. He holds over 50 patents. Jim received his Ph.D. from the University of Massachusetts (Amherst). He holds two M.A. degrees from the University of Utah.
57. Dan Wallach. Dan Wallach is a professor in the Department of Computer Science and a Rice Scholar at the Baker Institute for Public Policy at Rice University in Houston, Texas. His research considers a variety of different computer security topics, ranging

from web browsers and servers through Java security, electronic voting technologies, and smartphones. Wallach is a former member of the Air Force Science Advisory Board and a former member of the USENIX Association Board of Directors.

58. Peter Weinberger.\* Peter Weinberger is a computer scientist at Google. Previously, he was Chief Technology Officer at Renaissance Technologies and held many positions at Bell Labs, including Information Sciences Research Vice President where he was responsible for computer science research, math and statistics, and speech. As a scientist at Bell Labs he worked on Unix, contributing to the design and implementation of the AWK programming language, the IO library for f77, the fast factoring program qfactor, the B-tree library cbt, a code generator for C, and a network file system. He did research on topics including operating systems, compilers, security, and number theory. Before joining Bell Labs, he taught in the Department of Mathematics at the University of Michigan, Ann Arbor. He is a Fellow of the American Association for the Advancement of Science. He holds a B.S. from Swarthmore and a Ph.D. from the University of California, Berkeley.
59. Steve Wozniak. Steve Wozniak co-founded Apple and invented the Apple I and Apple II computers. He holds a B.S. in Electrical Engineering and Computer Science from UC Berkeley, and honorary doctorates from twelve universities. Wozniak is Innovator in Residence at High Point University. He founded many companies including CL 9, which brought the first programmable universal remote control to market in 1987, Wheels of Zeus (WOZ), and Acquicor Technology. He was Chief Scientist at Fusion-io and at Primary Data. He designed calculators for Hewlett-Packard and taught computer science to elementary school students and their teachers. Wozniak won numerous awards including the ACM Grace Murray Hopper Award, the National Medal of Technology (with Steve Jobs), the IEEE Hoover Medal, the Heinz Award for Technology, the American Humanist Association Isaac Asimov Science Award, the Global Award of the President of Armenia for Outstanding Contribution to Humanity Through IT, the Young Presidents' Organization Lifetime Achievement Award, the Cal Alumni Association Alumnus of the Year Award, and the Legacy for

Children Award from the Children's Discovery Museum in San Jose. He was named a Fellow of the Computer History Museum "for co-founding Apple Computer and inventing the Apple I personal computer," and inducted into the National Inventors Hall of Fame, the Manufacturing Hall of Fame, and the Consumer Electronics Hall of Fame.

60. Frank Yellin.\* Frank Yellin has spent over a decade working on runtime systems for interpreted and compiled languages. As a Staff Engineer in Embedded and Consumer at Sun Microsystems, he was an original member of the Java project. Yellin is co-author of *The Java Virtual Machine Specification* (Addison-Wesley, 1999), and co-authored the first version of the Java API specification. Previously he worked at Lucid, where he focused on multitasking, garbage collection, interrupts, and the compilation of Common Lisp. Yellin currently is a Staff Software Engineer at Google, where he works on automatic scalable security testing. He holds an A.B. in Applied Mathematics from Harvard and an M.S. in Computer Science from Stanford. He is the inventor or co-inventor of sixteen patents.