

1 KEKER & VAN NEST LLP  
ROBERT A. VAN NEST - # 84065  
2 [rvannest@kvn.com](mailto:rvannest@kvn.com)  
CHRISTA M. ANDERSON - # 184325  
3 [canderson@kvn.com](mailto:canderson@kvn.com)  
MICHAEL S. KWUN - # 198945  
4 [mkwun@kvn.com](mailto:mkwun@kvn.com)  
633 Battery Street  
5 San Francisco, CA 94111-1809  
Tel: 415.391.5400  
6 Fax: 415.397.7188

KING & SPALDING LLP  
DONALD F. ZIMMER, JR. - #112279  
[fzimmer@kslaw.com](mailto:fzimmer@kslaw.com)  
CHERYL A. SABNIS - #224323  
[csabnis@kslaw.com](mailto:csabnis@kslaw.com)  
101 Second Street, Suite 2300  
San Francisco, CA 94105  
Tel: 415.318.1200  
Fax: 415.318.1300

7 KING & SPALDING LLP  
SCOTT T. WEINGAERTNER  
8 (*Pro Hac Vice*)  
[sweingaertner@kslaw.com](mailto:sweingaertner@kslaw.com)  
9 ROBERT F. PERRY  
[rperry@kslaw.com](mailto:rperry@kslaw.com)  
10 BRUCE W. BABER (*Pro Hac Vice*)  
1185 Avenue of the Americas  
11 New York, NY 10036  
Tel: 212.556.2100  
12 Fax: 212.556.2222

IAN C. BALLON - #141819  
[ballon@gtlaw.com](mailto:ballon@gtlaw.com)  
HEATHER MEEKER - #172148  
[meeckerh@gtlaw.com](mailto:meeckerh@gtlaw.com)  
GREENBERG TRAURIG, LLP  
1900 University Avenue  
East Palo Alto, CA 94303  
Tel: 650.328.8500  
Fax: 650.328.8508

13 Attorneys for Defendant  
14 GOOGLE INC.

15 UNITED STATES DISTRICT COURT  
16 NORTHERN DISTRICT OF CALIFORNIA  
17 SAN FRANCISCO DIVISION

18 ORACLE AMERICA, INC.,  
19 Plaintiff,  
20 v.  
21 GOOGLE INC.,  
22 Defendant.

Case No. 3:10-cv-03561 WHA  
**GOOGLE'S 4/12/12 COPYRIGHT LIABILITY TRIAL BRIEF**  
Dept.: Courtroom 8, 19<sup>th</sup> Floor  
Judge: Hon. William Alsup

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

**TABLE OF CONTENTS**

	<u>Page</u>
I. INTRODUCTION .....	1
II. ARGUMENT .....	1
A. Computer programming languages are not copyrightable.....	1
1. Computer programming languages are systems for expression, or methods of operation for communication. ....	1
2. By the same token, the APIs are not copyrightable.....	3
3. Google has never taken the position that a computer programming language can be copyrighted.....	6
B. The APIs are integral to the Java programming language.....	7
1. Without the APIs, the Java programming language is deaf, dumb and blind.....	7
2. The APIs are fundamental to the Java programming language. ....	8
III. CONCLUSION.....	10

**TABLE OF AUTHORITIES**

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

**Page(s)**

**Federal Cases**

*Anti-Monopoly, Inc. v. General Mills Fun Group*  
611 F.2d 296 (9th Cir. 1979) ..... 5

*Baker v. Selden*  
101 U.S. 99 (1879)..... 2, 3, 4, 5, 6

*Chamberlin v. Uris Sales Corp.*  
150 F.2d 512 (2d Cir. 1945)..... 5

*Mazer v. Stein*  
347 U.S. 201 (1954)..... 6

*Publications Int’l v. Meredith Corp.*  
88 F.3d 473 (7th Cir. 1996) ..... 2

*Sega Enters. Ltd. v. Accolade, Inc.*  
977 F.2d 1510 (9th Cir. 1992) ..... 6

*Whist Club v. Foster*  
42 F.2d 782 (S.D.N.Y. 1929)..... 5

**Other Cases**

*SAS Institute Inc. v. World Programming Ltd.*  
Advocate General’s Opinion, Case C-406/10 (Nov. 29, 2011) ..... 3, 4

**Federal Statutes**

17 U.S.C. § 101 ..... 1

17 U.S.C. § 102(b) ..... 1, 2, 3, 5

**Federal Regulations**

37 C.F.R. § 202.1(c)..... 4

**Other Authorities**

Guy Steele, *Growing a Language* (Sun Microsystems, Oct. 1998)..... 1, 3, 9

1 *First Circuit Lotus v. Borland decision supports interoperability*, Business Wire,  
 Mar. 10, 1995 ..... 6

2

3 Brief for Amici Curiae of American Committee for Interoperable Systems and  
 Computer & Communications Industry Association in Support of Respondent,  
 4 *Lotus Development Corp. v. Borland International, Inc.*, 516 U.S. 233 (1996)  
 (No. 94-2003), 1995 WL 728487 ..... 6, 7

5 JOHN C. MITCHELL, *CONCEPTS IN PROGRAMMING LANGUAGES* (Cambridge Univ.  
 Press, 2003) (Trial Ex. 2507) ..... 2, 3

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

1 **I. INTRODUCTION**

2 The Court has asked for a “firm yes or no position on whether computer programming  
3 languages are copyrightable.” Order [Dkt 874] at 1. No, computer programming languages are  
4 not copyrightable. Google has never taken any other position. In addition, as requested, Google  
5 offers below a summary of some of the evidence it intends to present at trial relating to the  
6 copyrightability issues the Court has identified. *See* Order [Dkt. 865] at 1.

7 **II. ARGUMENT**

8 **A. Computer programming languages are not copyrightable.**

9 The Copyright Act defines a computer program as “a set of statements or instructions to  
10 be used directly or indirectly in a computer in order to bring about a certain result.” 17 U.S.C.  
11 § 101. A computer programming language is thus simply a language one can use to create a set  
12 of statements or instructions to be used directly or indirectly in a computer in order to bring about  
13 a certain result.<sup>1</sup> Without a computer programming language, the set of statements or instructions  
14 cannot be understood by the computer. As such, a computer language is inherently a utilitarian,  
15 nonprotectable means by which computers operate.

16 This commonsense approach to the statute makes the very distinction Congress itself  
17 drew: the protectable material is the computer program (the set of statements or instructions); the  
18 unprotectable material is the method or system (the language). So understood, original computer  
19 programs may be protected, but the medium for expression in which they are created is not.

20 **1. Computer programming languages are systems for expression, or  
21 methods of operation for communication.**

22 The Copyright Act bars copyright protection for an “idea, procedure, process, system,  
23 method of operation, concept, principle, or discovery,” even if it is in an “original work of  
24 authorship.” 17 U.S.C. § 102(b). This is what the Supreme Court meant when it stated that “no  
25 one would contend that the copyright of the treatise would give the exclusive right to the art or  
26 manufacture described therein.” *Baker v. Selden*, 101 U.S. 99, 102 (1879); *see also Publications*

27 <sup>1</sup> Guy Steele, an early member of the Java team at Sun, and now an Oracle Software Architect,  
28 defines a language as “a vocabulary and rules for what a string of words might mean to a person  
or a machine that hears them.” Guy Steele, *Growing a Language* (Sun Microsystems, Oct. 1998)  
 (“Steele”) at 2, available at <http://labs.oracle.com/features/tenyears/volcd/papers/14Steele.pdf>.

1 *Int'l v. Meredith Corp.*, 88 F.3d 473, 481 (7th Cir. 1996) (“The recipes at issue here describe a  
 2 procedure by which the reader may produce many dishes featuring Dannon yogurt. As such, they  
 3 are excluded from copyright protection as either a ‘procedure, process, [or] system.’”) (quoting  
 4 17 U.S.C. § 102(b)).

5 In the case of computer programs, this means that a given set of statements or instructions  
 6 may be protected, but the protection does not extend to the method of operation or system—the  
 7 programming language—by which they are understood by the computer. In copyright terms, the  
 8 set of statements or instructions is the expression and the language used to make that expression  
 9 intelligible to the machine is the method of operation or system. *See Google 4/3/12 Br.* [Dkt.  
 10 852] at 6, 14 (explaining that the APIs are a “*system* that can be *used* to express,” and that  
 11 computer languages are uncopyrightable for the same reason).<sup>2</sup> In its reply brief, Oracle did  
 12 not—and could not—dispute this point. *See Oracle 4/5/12 Br.* [Dkt. 859] at 3-4, 8. Oracle’s  
 13 expert, too, agrees: “Programming languages are the *medium of expression* in the art of computer  
 14 programming.” JOHN C. MITCHELL, *CONCEPTS IN PROGRAMMING LANGUAGES* (Cambridge Univ.  
 15 Press, 2003), Trial Ex. 2507 at 3 (emphasis added).

16 Oracle has no response to the common sense conclusion that a computer language is a  
 17 system for expression, except to argue that Section 102(b) must mean something else when it says  
 18 “system.”<sup>3</sup> Oracle’s own expert, however, has described programming languages as *abstractions*.

19 <sup>2</sup> Similarly, fictional languages such as Na’vi and Dothraki cannot be copyrighted. While the  
 20 film *Avatar* and the television series *Game of Thrones* are copyrightable (including the portions in  
 21 the fictional Na’vi and Dothraki languages), and while, for example, a dictionary or grammar  
 22 textbook for either language would be copyrightable, the languages *themselves* are not. Oracle  
 23 asks why copyright should not protect such languages, *see Oracle 4/5/12 Br.* [Dkt. 859] at 9; the  
 24 answer is that Section 102(b) says that they are not protected. Moreover, there is no reason to  
 25 believe that allowing copyright owners to control who can express themselves in these languages  
 26 would further the aims of copyright law.

27 <sup>3</sup> Oracle also argues that a computer language can be “original, text-based, and capable of  
 28 fixation,” and thus that it must be copyrightable. *See Oracle 4/5/12 Br.* [Dkt. 859] at 9. First,  
 Section 102(b) bars copyright protection for “original works of authorship” that fall within its  
 enumerated classes of exclusion. *See* 17 U.S.C. § 102(b). Thus, the fact that a system is original,  
 text-based and fixed does not mean that Section 102(b) does not apply.

Second, a language cannot be fixed. Certainly, a *description* of a language (e.g., a specification)  
 can be fixed. A computer program written *using* the language (e.g., the Gmail application on  
 Android phones) or an *implementation* of a language (e.g., a compiler or interpreter) can be fixed.  
 But none of those things *is* “the language,” any more than a dictionary “is” English, *Das Boot* “is”  
 German, or a C compiler “is” the C programming language. *See Baker*, 101 U.S. at 102 (“But

1 *See id.* at ix (“Programming languages provide the *abstractions*, organizing *principles*, and  
 2 control structures that programmers use to write good programs.”) (emphases added).<sup>4</sup> Thus even  
 3 Oracle’s own expert places programming languages firmly on the unprotectable “idea” side of the  
 4 idea/expression dichotomy.

5 **2. By the same token, the APIs are not copyrightable.**

6 The APIs at issue are integral to and part of the Java programming language. *See infra*  
 7 Part II.B. But whether that is the case, or whether they can be separated from the Java  
 8 programming language (as Oracle argues), it is undeniable that these APIs extend the language by  
 9 increasing its vocabulary. *See Steele* at 7 (“A true library does not change the rules of meaning  
 10 for the language; it just adds new words.”). Whether the collective set is called “the Java  
 11 programming language” (adopting Google’s view) or perhaps “the Java programming language  
 12 supercharged” (adopting Oracle’s view), it is, to use Guy Steele’s definition, “a vocabulary and  
 13 rules for what a string of words might mean to a person or a machine that hears them.” *Steele* at  
 14 2. The whole of this collective set is thus as uncopyrightable as any programming language. The  
 15 APIs, as a subset of this uncopyrightable whole, are themselves uncopyrightable. *See* 17 U.S.C.  
 16 § 102(b).

17 In its April 5th brief, Oracle argues that the opinion of the ECJ Advocate General suggests  
 18 that “interfaces” can be copyrighted, at least in some circumstances. *See* Advocate General’s  
 19 Opinion, *SAS Institute Inc. v. World Programming Ltd.*, Case C-406/10, ¶ 85 (Nov. 29, 2011).  
 20 The opinion, however, uses “interface” in two senses, first referring to a file format (which it  
 21 concludes is an uncopyrightable idea) and later referring to specific source code in a computer  
 22 program, authored by the developer, that *implements* a file format (which it concludes may be  
 23 copyrighted). This is entirely consistent with Google’s position.

24 there is a clear distinction between the book, as such, and the art which it is intended to illustrate.  
 25 The mere statement of the proposition is so evident, that it requires hardly any argument to  
 support it.”); *cf.* René Magritte, *La trahison des images*.

26 <sup>4</sup> Oracle’s expert has further described designing a programming language as requiring decisions  
 27 regarding what *ideas* to leave out. *See id.* at 3 (“A single application also helps with one of the  
 most difficult parts of language design: leaving good ideas out.”). And he has described studying  
 28 programming languages as requiring “the study of *conceptual* frameworks for problem solving,  
 software construction, and development.” *Id.* at 5 (emphasis added).

1           The opinion first calls the file format used by SAS for data files a “logic interface.” *See*  
2 *id.* ¶¶ 77-78. “Those formats may be regarded as blank forms which are to be filled with the  
3 customer’s data by the SAS System and which contain specific locations in which particular  
4 information must be written in order for the system to be read and write the file correctly.” *Id.*  
5 ¶ 79. Blank forms are *per se* uncopyrightable under the Copyright Act. *Baker*, 101 U.S. at 107;  
6 37 C.F.R. § 202.1(c).

7           Next, the opinion discusses how this logic interface—the file format—could be made part  
8 of a computer program, explaining that “interface” could also refer to “the elements which create,  
9 write and read the format of said SAS data files” which are “expressed in source code in the  
10 program.” *SAS*, Case C-406/10, op. at ¶ 82. The opinion concludes that the SAS source code that  
11 *implements* the file format could be protected by copyright. *See id.* ¶¶ 81, 82. The Advocate  
12 General opines that under EU law, WPL was nonetheless allowed to decompile this code to  
13 reverse engineer the file format, so long as WPL wrote its own code to implement the file format.  
14 *See id.* ¶¶ 83-90. In short, the opinion is consistent with Google’s view, and distinguishes  
15 between the *idea* represented by an interface, which is not copyrightable, and the *source code*  
16 *implementing* an interface, which may be protected by copyright.

17           That these APIs are an uncopyrightable idea, system or method of operation becomes  
18 clearer still when one focuses on precisely what Oracle claims is copyrightable: the structure,  
19 selection and organization of the APIs. A set of nonsensical APIs could be created that had  
20 exactly the same structure, selection and organization as the Oracle APIs, but that *did* different  
21 things. For example, the `sqrt()` method could always return zero—indeed, every method that  
22 returns a number could always return zero, while those that return text could always return the  
23 letter *a*, those that return true or false could always return *true*, and so on, with a default result  
24 being used for every variable type. This set of APIs would serve no useful purpose, but would  
25 have *exactly* the same structure, selection and organization as the Oracle APIs. No reasonable  
26 jury could ever conclude that the “expression” in this hypothetical set of APIs is substantially  
27 similar to the “expression” in the Oracle APIs, notwithstanding the “copied” structure, selection  
28 and organization. Thus, Oracle’s infringement theory fails unless it accuses not just the structure,

1 selection and organization, but also the *purpose* of the APIs. In other words, Oracle’s  
2 infringement claim fails unless it is allowed to copyright *ideas*, which it cannot do. 17 U.S.C.  
3 § 102(b); *see also Anti-Monopoly, Inc. v. General Mills Fun Group*, 611 F.2d 296, 300 n.1 (9th  
4 Cir. 1979) (“business ideas, such as a game concept, cannot be copyrighted”); *Chamberlin v. Uris*  
5 *Sales Corp.*, 150 F.2d 512, 513 (2d Cir. 1945) (“Precisely, however, because it is the form of  
6 expression and not the idea that is copyrightable, we hold that the defendant did not infringe on  
7 the plaintiff’s statement of the rules. The similarities of the two sets of rules derive from the fact  
8 that they were necessarily drawn from the same source.”); *Whist Club v. Foster*, 42 F.2d 782  
9 (S.D.N.Y. 1929) (“In the conventional laws or rules of a game, as distinguished from the forms or  
10 modes of expression in which they may be stated, there can be no literary property susceptible of  
11 copyright.”).

12 Indeed, Oracle now—on the eve of trial—candidly states that it claims Google’s  
13 *implementing source code* is a derivative work of *Oracle’s English-language descriptions*  
14 because Google’s source code *does the things that the English descriptions describe*. *See* Oracle  
15 4/5/12 Br. [Dkt. 859] at 10 (Oracle is claiming infringement based on “Google’s creation of  
16 derivative works from the English-language descriptions of the elements of the API  
17 specifications”). That is nothing but an assertion that *Google’s expression* infringes *Oracle’s*  
18 *ideas*. Oracle thus stands as an exception to the Supreme Court’s statement that “no one would  
19 contend that the copyright of the treatise would give the exclusive right to the art or manufacture  
20 described therein.” *Baker*, 101 U.S. at 102.

21 While Oracle argues the “extremity” of Google’s position, the truly extreme position  
22 would be to allow a party to devise a *system* (the Java language APIs), and then enforce  
23 copyrights in *descriptions* of that system (Oracle’s specifications) and *implementations*  
24 (*expressions*) of that system (Oracle’s libraries) to preclude others from *practicing* the system.  
25 Oracle’s approach is barred by *Baker v. Selden*:

26 To give to the author of the book an exclusive property in the art described therein,  
27 when no examination of its novelty has ever been officially made, would be a  
28 surprise and a fraud upon the public. That is the province of letters-patent, not of  
copyright. The claim to an invention or discovery of an art or manufacture must  
be subjected to the examination of the Patent Office before an exclusive right

1           therein can be obtained; and it can only be secured by a patent from the  
2           government.

3           101 U.S. at 102. It is barred by *Mazer v. Stein*: “Unlike a patent, a copyright gives no exclusive  
4           right to the art disclosed; protection is given only to the expression of the idea—not the idea  
5           itself.” 347 U.S. 201, 217 (1954). It is barred by *Sega Enters. Ltd. v. Accolade, Inc.*, under which  
6           “functional requirements for compatibility” with a system described by or implemented in a  
7           copyrighted work cannot be protected by copyright law. 977 F.2d 1510, 1522 (9th Cir. 1992).

8                           **3.       Google has never taken the position that a computer programming  
9                           language can be copyrighted.**

10           Google has never taken the position, before a court or agency or otherwise, that a  
11           programming language was or is copyrightable. Google does believe that computer source code  
12           *implementing* a language can be copyrighted. Google has, for example, created programming  
13           languages called “GO” and “Dart.” Google has encouraged others to use these languages for free,  
14           and has also provided an open source license for others to use Google’s *source code and object*  
15           *code* that *implements* these languages. This is consistent with the positions Google has taken in  
16           this case.

17           Google notes that Sun (now known as Oracle America) organized, formed and led the  
18           American Committee for Interoperable Systems (“ACIS”),<sup>5</sup> the chairperson of which was Sun’s  
19           Deputy General Counsel, Peter M.C. Choy. In a press release after the First Circuit’s decision in  
20           *Lotus v. Borland*, Mr. Choy “noted that the decision will make it more difficult for vendors to  
21           attempt to lock out competitors and lock in consumers by asserting proprietary rights in certain  
22           ‘building blocks’ of software, *such as programming languages, program interfaces, and the*  
23           *functional aspects of user interfaces.*” *First Circuit Lotus v. Borland decision supports*  
24           *interoperability*, Business Wire, Mar. 10, 1995 (emphasis added).<sup>6</sup> Mr. Choy was also counsel of  
25           record for an ACIS *amicus* brief filed with the Supreme Court, urging the Court to affirm the First  
26           Circuit’s judgment that the Lotus menu hierarchy was not copyrightable. ACIS argued that “[t]he

26           <sup>5</sup> The organization had the same mailing address as Sun’s headquarters. At the time of the *Lotus*  
27           case, the ACIS website was located at <http://www.sun.com/ACIS/>.

28           <sup>6</sup> Available via LEXIS-NEXIS. Sun also distributed this press release by other means. *See, e.g.*,  
<http://www3.wcl.american.edu/cni/9503/4860.html>.

1 decisive issue in [the *Lotus*] case is whether copyright law can protect the rules that enable two  
2 elements of a computer system to work together.” 1995 WL 728487, at \*3. ACIS further argued:

3 The 1-2-3 command structure is more than a user interface; it is the interface  
4 between the Lotus program and *programs—referred to as “macros”*—that are  
5 written by users at their own considerable expense for execution in connection  
6 with the 1-2-3 program. Because *the 1-2-3 command structure provides the  
7 template for the macros* and because the macros are the key to compatibility, the  
8 First Circuit, consistent with holdings in other circuits, ruled that those elements  
9 necessary to macro compatibility are not protected by copyright.

10 *Id.* (emphases added). Thus, while not directly taking a position on whether programming  
11 languages can be copyrighted, the brief implies that they cannot.

12 **B. The APIs are integral to the Java programming language.**

13 As Google has previously noted, Java’s own books describing the APIs state that they are  
14 available “to all Java programs . . . .” Trial Ex. 980 at xviii. Those books describe four of the  
15 APIs (out of eight that then existed) as “the foundation of the Java language.” *Id.* (back cover).

16 **1. Without the APIs, the Java programming language is deaf, dumb and  
17 blind.**

18 The APIs are so fundamental that without them the Java programming language has no  
19 ability to provide any output to the user. Similarly, without the APIs, the Java programming  
20 language has no ability to accept input from the user.<sup>7</sup> When Mark Reinhold, Oracle’s Chief  
21 Architect of the Java Platform, was asked why the Java language APIs exist, he testified:

22 Well, if there were no APIs, we would only have a language. You would be able  
23 to write basic computations that never did any IO, had any communication with  
24 the outside world or the underlying platform.

25 You could write—you know, you could do computations on numbers and  
26 strings and generate them, but you wouldn’t be able to do anything with them.

27 Reinhold 8/5/11 Depo. at 115:10-17.<sup>8</sup> He further explained, “But even doing that, even just to  
28 manipulate a string requires the string API, so you’re—you’re, actually, pretty much just limited  
to numbers, which are pretty boring.” *Id.* at 115:19-22.

<sup>7</sup> There is one minor exception. A Java language program can be written to accept arguments  
from the “command line” at runtime. Even this facility, however, is limited to accepting a *single  
set* of arguments at the *beginning* of the program.

<sup>8</sup> This testimony is subject to an objection, but only that the testimony is outside the scope of the  
Rule 30(b)(6) topics for which Dr. Reinhold was designated.

1 In this respect, the Java language APIs are similar to libraries associated with some older  
 2 languages in the history of programming. In C, for example, input and output facilities are part of  
 3 what the designers of the C language called “the standard library, a set of functions that provide  
 4 input, output, string handling, storage management, mathematical routines, and a variety of other  
 5 services for C programs.” BRIAN W. KERNIGHAN & DENNIS M. RITCHIE, THE C PROGRAMMING  
 6 LANGUAGE (Prentice Hall, 2d ed., 1988), Trial Ex. 3002 at 151.<sup>9</sup> Even the basic “hello, world”<sup>10</sup>  
 7 C program in their book requires using the standard library in order to display the words “hello,  
 8 world” to the user. *See id.* at 6.<sup>11</sup> Similarly, Oracle’s “hello, world” program in the Java  
 9 programming language includes the following source code:

10 System.out.println("Hello World!");<sup>12</sup>

11 “System” refers to a class that is part of the java.lang API package, and “out” is a field defined in  
 12 the System class. The System class defines the “out” field as belonging to the “PrintStream”  
 13 class, which is part of the java.io API package. Thus, even implementing this most basic of  
 14 programs in the Java programming language requires using two of the accused APIs.

## 15 2. The APIs are fundamental to the Java programming language.

16 In its April 5th brief, Oracle conceded that the Java language specification *requires* the  
 17 defineClass() method from the ClassLoader class in the java.lang package. *See* Oracle 4/5/12 Br.  
 18 [Dkt. 859] at 7. In J2SE 5.0, the defineClass() method is an “overloaded” method; there are four  
 19 versions of the defineClass() method, the fourth of which has the following method declaration:

20 protected Class defineClass(String name, ByteBuffer b, ProtectionDomain  
 21 protection Domain)

22 <sup>9</sup> The “Standard Template Library” is a similar library that has been incorporated into the  
 23 standard C++ specification.

24 <sup>10</sup> The authors explain that a “hello, world” program—a program that prints the words “hello,  
 world”—is typically the first program a developer writes when learning a language. *See id.* at 5.

25 <sup>11</sup> The authors state that “[i]nput and output facilities,” which are part of the standard library, “are  
 26 not part of the C language itself . . . .” *Id.* 151. Even the basic “hello, world” C program they  
 27 introduce, however, requires the standard library. All that is meant by their distinction between  
 the “language” and the library is that “higher-level mechanisms must be provided in explicitly-  
 called functions.” *Id.* at 2. That is, they require APIs. Notably, the authors discuss the C  
 standard library as part of their book about the “C programming language.”

28 <sup>12</sup> *See* <http://docs.oracle.com/javase/tutorial/getStarted/application/index.html>.

1 As indicated in the parentheses, the method accepts three arguments, of types “String,”  
2 “ByteBuffer” and “ProtectionDomain.” String, ByteBuffer and ProtectionDomain are classes  
3 defined, respectively, in the java.lang, java.nio and java.security APIs. Implementing this *single*  
4 *example of a single required class* thus requires implementing elements of *three* of the 37 APIs.<sup>13</sup>

5 This is only a single example—a single example that *Oracle* chose to highlight. Due to  
6 the interdependencies between classes in the APIs, expressly requiring one element often will  
7 necessarily require many others, just as the `defineClass()` method implicates the String,  
8 ByteBuffer and ProtectionDomain classes from java.lang, java.nio<sup>14</sup> and java.security. Based on  
9 the classes expressly required by the Java language specification and interdependencies in the  
10 APIs, thousands of elements from the accused APIs are required in order to implement the Java  
11 programming language.<sup>15</sup> In fact, the first edition of the Java language specification devotes over  
12 300 pages to documentation for the java.lang, java.io and java.net packages. *See* Trial Ex. 4027  
13 at 455-765. The documentation of the APIs was removed from later editions of the Java language  
14 specification only for space reasons. *See* Trial Ex. 984 at xxvi (“The specifications of the  
15 libraries are now far too large to fit into this volume, and they continue to evolve. Consequently,  
16 API specifications have been removed from this book.”).

17 <sup>13</sup> In its April 5th brief, Oracle also suggests that one could implement the `defineClass()` method  
18 *without* implementing the rest of the `ClassLoader` class. *See* Oracle 4/5/12 Br. [Dkt. 859] at 7.  
19 Oracle has repeatedly claimed that it is being irreparably harmed by alleged “fragmentation”  
20 because Google did not fully implement *all* of the J2SE API packages. Here, however, it appears  
21 to argue that to implement the free and open Java programming language, one should implement  
22 only *part* of the APIs. To the extent that Android “fragments” Java at all—and witnesses at trial  
23 will dispute this point—the approach Oracle appears to suggest would “fragment” Java far more.  
24 Further, the evidence at trial will show that Java, and particularly Java ME, was “fragmented”  
25 long before Android, and that Sun condoned this “fragmentation.”

22 <sup>14</sup> Oracle argues that because some of the accused packages were not part of the initial release of  
23 Java, they cannot be fundamental or integral to the Java programming language. Languages,  
24 however, are not static. *See* Trial Ex. 984 at xxv (“This specification defines the language as it  
25 exists today. The language is likely to continue to evolve.”); *Steele* at 5 (“I now think that I, as a  
language designer who helps out with the design of the Java programming language, need to ask  
not ‘Should the Java programming language grow?’ but ‘How should the Java programming  
language grow?’”).

26 <sup>15</sup> Oracle argues that when the Java language specification refers to APIs that are fully defined  
27 elsewhere, that means that the referenced definitions are not part of the language. *See* Oracle  
28 4/12/12 Br. [Dkt 859] at 7. This is backwards. When the Java language specification “does not  
provide a complete specification” but refers the reader to the APIs for details, *see* Trial Ex. 984 at  
6, the only fair conclusion is that the language specification is incorporating material by reference  
from the API specifications.

1 In addition, witnesses at trial will testify that developers expect the APIs to be available  
 2 when they program in the Java programming language, that the APIs are routinely taught in  
 3 beginning courses regarding use of the language, and that no developer can credibly claim to be  
 4 proficient in the Java programming language unless he or she knows the APIs. And, in addition  
 5 to statements highlighted in prior briefs, Sun also stated, for example, that the java.lang API  
 6 “provides the classes and interfaces that form the core of the Java language and the Java Virtual  
 7 Machine,” and that several objects defined in java.lang are “closely intertwined with the Java  
 8 language definition.” Trial Ex. 980 at xix. Oracle’s expert has testified that the Java  
 9 programming language cannot be implemented without including at least some of the APIs.

10 Indeed, Sun described the Java programming language as follows:

11 *The Java programming language* is a general-purpose concurrent class-based  
 12 object-oriented programming language, specifically designed to have as few  
 13 implementation dependencies as possible. It *allows application developers to*  
 14 *write a program once and then be able to run it everywhere on the Internet.*

14 Trial Ex. 984 at xxi (emphasis added). Because any useful program in the Java programming  
 15 language requires the APIs, “the Java programming language” only allows a developer to write a  
 16 program once and run it everywhere if the “language” is understood to include the APIs.

### 17 III. CONCLUSION

18 Computer programming languages are not copyrightable, and neither are Oracle’s APIs.  
 19 Oracle accuses Google of infringement for *doing* what the Oracle API specifications *describe*.  
 20 That is a classic attempt to improperly assert copyright over an *idea* rather than *expression*. The  
 21 Court should hold that the structure, selection and organization of the APIs are uncopyrightable.

22 Dated: April 12, 2012

KEKER & VAN NEST LLP

23 By: /s/ Robert A. Van Nest  
 ROBERT A. VAN NEST

24 Attorneys for Defendant  
 25 GOOGLE INC.