

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

7 KING & SPALDING LLP
8 SCOTT T. WEINGAERTNER
(*Pro Hac Vice*)
sweingaertner@kslaw.com
9 ROBERT F. PERRY
rpperry@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

13 Attorneys for Defendant
14 GOOGLE INC.

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

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

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-03651 WHA
GOOGLE'S 4/3/12 COPYRIGHT LIABILITY TRIAL BRIEF
Dept.: Courtroom 8, 19th Floor
Judge: Hon. William Alsup

TABLE OF CONTENTS

Page

	I.	INTRODUCTION	1
	II.	ARGUMENT	1
	A.	There is no compilation exception to the prohibition against copyrighting ideas, systems and methods of operation.....	1
	B.	The CONTU Report does not directly address the applicability of Section 102(b), but leaves that determination to development through court decisions.....	3
	C.	The accused aspects of the API specifications are an uncopyrightable system or method of operation.....	4
	D.	The merger doctrine is not limited to high levels of abstraction, and bars copyright protection for any arguable expression in the selection, arrangement and structure of the APIs.....	8
	E.	The Java language APIs at issue are deeply intertwined with and integral to the Java programming language.	11
	1.	More than half of the accused API packages are described by Sun as “fundamental” to the Java programming language.	11
	2.	The other 18 accused API packages relate to features that are standard in modern applications.	12
	3.	The 37 Java language API packages are all integral to the Java programming language.	13
	F.	Computer languages are neither copyrightable nor patentable.....	14
	1.	Computer languages are not copyrightable.....	14
	2.	Computer languages are not patentable.	16
	G.	Subject to the requirements of the Patent Act, it may be possible to try to claim the selection of classes for APIs under patent law.....	18
	H.	Aside from its “selection, arrangement and structure” argument, Oracle’s only argument that Google’s <i>implementation</i> of the APIs infringes is based on nine lines of mundane code from Arrays.java.	19
	III.	CONCLUSION.....	20

TABLE OF AUTHORITIES

Federal Cases

4	<i>Allen v. Academic Games League of America, Inc.</i>	9
	89 F.3d 614 (9th Cir. 1996)	
5	<i>Apple Computer, Inc. v. Microsoft Corp.</i>	
	35 F.3d 1435 (9th Cir. 1994)	3
6	<i>ATC Distribution Group, Inc. v. Whatever It Takes Transmissions & Parts, Inc.</i>	
	402 F.3d 700 (6th Cir. 2005)	2
7	<i>Baker v. Selden</i>	
	101 U.S. 99 (1879).....	18, 19
8	<i>Bilski v. Kappos</i>	
	130 S. Ct. 3218 (2010).....	16
9	<i>Engineering Dynamics, Inc. v. Structural Software, Inc. (“EDI I”)</i>	
	26 F.3d 1335 (5th Cir. 1994)	9
10	<i>Engineering Dynamics, Inc. v. Structural Software, Inc. (“EDI II”)</i>	
	46 F.3d 408 (5th Cir. 1995)	9, 10
11	<i>Feist Publications, Inc. v. Rural Telephone Service Co., Inc.</i>	
	499 U.S. 340 (1991).....	2, 7
12	<i>Gates Rubber Co. v. Bando Chem. Indus., Ltd.</i>	
	9 F.3d 823 (10th Cir. 1993)	2
13	<i>In re Alappat</i>	
	33 F.3d 1526 (Fed. Cir. 1994) (en banc).....	17
14	<i>In re Comiskey</i>	
	554 F.3d 967 (Fed. Cir. 2009).....	17
15	<i>In re Ferguson</i>	
	558 F.3d 1359 (Fed. Cir. 2009).....	17
16	<i>Johnson Controls, Inc. v. Phoenix Control Sys., Inc.</i>	
	886 F.2d 1173 (9th Cir. 1989)	7
17	<i>Lotus Dev. Corp. v. Borland Int’l, Inc.</i>	
	49 F.3d 807 (1st Cir. 1995)	8
18	<i>Matthew Bender & Co., Inc. v. West Pub. Co.</i>	
	158 F.3d 674 (2d Cir. 1998).....	2
19	<i>Nichols v. Universal Pictures Corp.</i>	
	45 F.2d 119 (2d Cir. 1930).....	2

1	<i>Sega Enters. Ltd. v. Accolade, Inc.</i>	10
2	977 F.2d 1510 (9th Cir. 1992)	

Other Cases

3	<i>Advocate General's Opinion, SAS Institute Inc. v. World Programming Ltd.</i>	
4	Case C-406/10 (Nov. 29, 2011)	16
5	<i>SAS Institute, Inc. v. World Programming Ltd.</i>	
6	[2010] EWHC (Ch) 1829 (Eng.).....	14, 15

Federal Statutes

9	17 U.S.C. § 101.....	<i>passim</i>
10	17 U.S.C. § 102(a)	1
11	17 U.S.C. § 102(b).....	<i>passim</i>
12	35 U.S.C. § 102.....	18
13	35 U.S.C. § 103.....	18

Other Authorities

16	Guy L. Steele, <i>Growing a Language</i> (Sun Microsystems, Oct. 1998)	13
17	JONATHAN BAND & MASANOBU KATOH, <i>INTERFACES ON TRIAL 2.0</i> (MIT Press 2011)	3
18	Pamela Samuelson, <i>Why Copyright Law Excludes Systems and Processes from the Scope of Protection</i> , 85 TEX. L.R. 1921 (2007).....	3, 5, 7
19	Richard H. Stern, <i>Copyright in Computer Programming Languages</i> , 17 RUTGERS COMPUTER & TECH L.J. 321 (1991).....	16
20	Sebastian Zimmeck, <i>Patent Eligibility of Programming Languages and Tools</i> , 13 TUL. J. TECH. & INTELL. PROP. 133 (2010)	17

1 **I. INTRODUCTION**

2 Based on the copyright liability trial briefs filed by the parties to date, and in light of the
 3 answers given below to the questions posed by the Court in its March 29, 2012 Order, the
 4 selection, arrangement and structure of the 37 APIs are not copyrightable.

5 First, the selection, arrangement and structure of the APIs are part of a medium for
 6 expression, not the expression itself, and as such are an uncopyrightable system or method of
 7 operation. Second, the merger doctrine bars copyright protection for any arguable expression in
 8 the selection, arrangement and structure of the APIs. Third, Oracle has repeatedly conceded that
 9 it is not asserting copyright over the Java programming language itself, and yet the APIs at issue
 10 are integral to that language. Finally, allowing copyright protection for the selection,
 11 arrangement and structure of the APIs would be no different than allowing Oracle to copyright a
 12 programming language, which is barred by 17 U.S.C. § 102(b).

13 **II. ARGUMENT**

14 **A. There is no compilation exception to the prohibition against copyrighting
 15 ideas, systems and methods of operation.**

16 The Court has asked three questions addressing the role of selection, arrangement and
 17 structure in copyright law. *See* 3/29/12 Order, Questions 1-3. Selection, arrangement and
 18 structure allows for the *possibility* of copyright protection for non-literal elements of a work.
 19 Such possible protection, however, is always subject to Section 102(b) of the Copyright Act.
 20 There is no compilation exception to the prohibition against copyrighting ideas, systems and
 21 methods of operation.

22 Copyright protects “original works of authorship.” 17 U.S.C. § 102(a).¹ However, “[i]n
 23 *no case* does copyright protection for an original work of authorship extend to any *idea*,
 24 *procedure, process, system, method of operation, concept, principle, or discovery, regardless of*
 25 *the form* in which it is described, explained, illustrated, or embodied in such work.” *Id.* § 102(b)
 26 (emphases added). Thus, in answer to Question 2 from the Court’s March 29, 2012 Order [Dkt.

27 ¹ There are other requirements as well. For example, the work must be fixed, *see id.*, and the
 28 copyright owner’s rights are expressly limited by numerous sections of the Copyright Act,
 including fair use, *see id.* § 106 (exclusive rights are “[s]ubject to sections 107 through 122” of
 the Copyright Act).

1 838], by the express terms of Section 102(b), there are *no* exceptions to the rule that systems and
 2 methods of operation are not copyrightable. Moreover, Section 102(b) excludes protection for *all*
 3 systems and methods of operation, without regard for whether they are original, creative, elegant,
 4 life-changing or difficult to develop.²

5 The Court and the parties have referenced the “selection, arrangement and structure” of
 6 the elements of the APIs. Other courts have used the alternative formulation “structure, sequence
 7 and organization.” *See, e.g., Gates Rubber Co. v. Bando Chem. Indus., Ltd.*, 9 F.3d 823, 836
 8 (10th Cir. 1993). The Copyright Act uses similar language to define a “compilation.”
 9 Specifically, the Act defines a “compilation” as:

10 a work formed by the collection and assembling of preexisting materials or of data
 11 that are *selected, coordinated, or arranged* in such a way that the resulting work
 as a whole constitutes an original work of authorship.

12 17 U.S.C. § 101 (emphasis added). Thus, as the Court suggests, even where the elements of a
 13 work are noncopyrightable, it is *possible* that the selection, coordination and arrangement of those
 14 elements can be copyrighted. *See* 3/29/12 Order, Question 1. That said, not *every* selection,
 15 coordination and arrangement of elements can be copyrighted—it is copyrightable *only if* “the
 16 resulting work as a whole constitutes an original work of authorship.” 17 U.S.C. § 101.³

17 ² *See Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 122 (2d Cir. 1930) (Learned Hand, J.)
 18 (“Even so, granting that the plaintiff’s play was wholly original, and assuming that novelty is not
 19 essential to a copyright, there is no monopoly in such a background. Though the plaintiff
 20 discovered the vein, she could not keep it to herself; so defined, the theme was too generalized an
 abstraction from what she wrote. It was only a part of her ‘ideas.’”); *ATC Distribution Group,*
 21 *Inc. v. Whatever It Takes Transmissions & Parts, Inc.*, 402 F.3d 700, 707 (6th Cir. 2005)
 22 (“Original and creative *ideas*, however, are not copyrightable, because 17 U.S.C. § 102(b)
 provides that ‘in no case does copyright protection for an original work of authorship extend to
 any idea, procedure, process, system, method of operation, concept, principle, or discovery,
 regardless of [its] form.’”).

23 ³ Moreover, to constitute an “original work of authorship” a compilation of otherwise
 24 unprotectable elements (such as the API names and elements at issue here) must exhibit creativity
 25 in the selection, coordination and arrangement of its elements. *See Feist Publications, Inc. v.*
 26 *Rural Telephone Service Co., Inc.*, 499 U.S. 340 (1991) (rejecting “sweat of the brow” doctrine).
 27 As explained by the Second Circuit, “when it comes to the selection or arrangement of
 information, creativity inheres in making non-obvious choices from among more than a few
 options.” *Matthew Bender & Co., Inc. v. West Pub. Co.*, 158 F.3d 674, 682 (2d Cir. 1998). To
 28 the extent that Oracle claims copyright protection in a compilation, the relevant compilation
 would be the one that Oracle actually originated—the full compilation of J2SE 5.0, which
 includes among other things 166 APIs, not merely the accused subset of 37 APIs. Because
 Oracle has not registered the works as compilations, Oracle bears the burden of proving both that
 there is sufficient creativity in the selection, arrangement and organization of the entire

1 A compilation, like any original work of authorship, is subject to Section 102(b), because
 2 “[i]n no case does copyright protection for an original work of authorship extend to” ideas,
 3 systems or methods of operation. *Id.* § 102(b) (emphasis added). Thus, if a compilation is itself
 4 a system or method of operation, copyright cannot protect it. *See id.* Not only is this compelled
 5 by the language of the statute (“[i]n no case”), but as a matter of logic. As the Court noted,
 6 “original methods and systems” will, by definition, have a “structure, arrangement and selection,”
 7 and thus if having a structure, arrangement and selection served as an exception to the rule against
 8 copyrighting methods or systems, that would render much of Section 102(b) a dead letter. *See*
 9 3/29/12 Order [Dkt. 838], Question 3. In sum, the exclusion in Section 102(b) is absolute, and
 10 applies regardless of whether the system or method of operation is an “original” compilation.

11 **B. The CONTU Report does not directly address the applicability of Section
 12 102(b), but leaves that determination to development through court decisions.**

13 The Final Report of the Commission on New Technological Uses of Copyrighted Works
 14 (“CONTU Report”) recommended that Congress “make it explicit that computer programs, to the
 15 extent that they embody an author’s original creation, are proper subject matter of copyright.”
 16 CONTU Report at 1. Congress adopted this recommendation. *See* Pamela Samuelson, *Why*
 17 *Copyright Law Excludes Systems and Processes from the Scope of Protection*, 85 TEX. L.R. 1921,
 18 1954 n.212 (2007) (“Samuelson”)⁴ (“Congress followed CONTU’s recommendations by adding a
 19 definition of computer programs to the statute and amending § 117”).

20 The CONTU Report, however, had no clear recommendation about the extent to which
 21 non-literal aspects of a computer program might be copyrightable. The Report states that “one is
 22 always free to make a machine perform any conceivable process (in the absence of a patent), but

23 compilation of J2SE 5.0 to be entitled to protection, and that the accused subset of 37 APIs
 24 reflects the creative portion or is virtually identical to the entire compilation. *See Apple*
Computer, Inc. v. Microsoft Corp., 35 F.3d 1435, 1439 (9th Cir. 1994) (where “the range of
 25 protectable and unauthorized expression is narrow, the appropriate standard for illicit copying is
 virtual identity”).

26 ⁴ In addition to Professor Samuelson’s article, Google recommends that the Court consult
 27 JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL 2.0 (MIT Press 2011), *available at*
http://mitpress.mit.edu/books/full_pdfs/Interfaces_on_Trial_2.0.pdf (free download). This book
 28 updates a 1995 book by the same authors, and provides a comprehensive overview of domestic
 and foreign cases addressing interoperability. Chapters 1 and 2 are particularly relevant to the
 present case.

one is not free to take another's program." CONTU Report at 20. However, "copyright protection for programs does not threaten to block the use of ideas or program language previously developed by others when that use is necessary to achieve a certain result." *Id.* Statements such as these in the Report merely restate principles of law that were already well established, instead of proposing a new framework for analysis of computer programs.

Ultimately, the Report offers two uncontroversial propositions. First, photocopying a printed computer program listing is "clearly an infringement." *Id.* at 22. At the other end of the spectrum, "anyone is free to make a computer carry out any unpatented process, but not to misappropriate another's writing to do so." *Id.* Beyond these two clear-cut situations, the CONTU Report left the rest to the courts to decide on a case-by-case basis:

Drawing the line between the copyrightable form of a program and the uncopyrightable process which it implements is simple in the first instance described above. But the many ways in which programs are now used and the new applications which advancing technology will supply may make drawing the line of demarcation more and more difficult. To attempt to establish such a line in this report written in 1978 would be futile. Most infringements, at least in the immediate future, are likely to involve simply copying. In the event that future technology permits programs to be stated orally for direct input to a computer through auditory sensing devices or permits future infringers to use an author's program without copying, difficult questions will arise. *Should a line need to be drawn to exclude certain manifestations of programs from copyright, that line should be drawn on a case-by-case basis by the institution designed to make fine distinctions—the federal judiciary.*

Id. at 22-23 (emphasis added).

C. The accused aspects of the API specifications are an uncopyrightable system or method of operation.

The Copyright Act defines a "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101. The source code that implements, for example, the max() method that is part of the Math class in the java.lang package meets this definition—it is a set of instructions that is used by a computer to determine which of two numbers provided to it is larger. The source code that implements the getContent() method that is part of the URL class in the java.net package also meets this definition—it is a set of instructions that is used by a computer to retrieve the content at the website hosted at a given URL. But, leaving aside the nine-line rangeCheck() utility

1 method that is part of its literal copying claims,⁵ Oracle doesn't claim that the source code
 2 implementing any of the API elements is copied.

3 Instead, Oracle relies on alleged copying at a higher level of abstraction—the selection,
 4 arrangement and structure of the 37 API subset of the 166 J2SE 5.0 APIs. In many computer
 5 programs, the non-literal structure of a program may still fit the statutory definition of a computer
 6 program. For example, a developer might create a flow chart for a video game program that
 7 describes, at a high level, the logic flow that the computer will follow when executing the video
 8 game application. Although not the literal instructions that the computer follows, the flow chart
 9 could still be thought of as a set of instructions that *indirectly* are used by a computer to bring
 10 about a certain result. Depending on the facts, this logic flow might be so general that it falls on
 11 the idea side of the idea/expression dichotomy. But if it falls on the expression side, it might be
 12 protected by copyright. In making that determination, courts must remember that the term "idea"
 13 in the phrase "idea/expression dichotomy" is drawn from Section 102(b), and should be
 14 understood to refer to all of the statutorily excluded classes listed in Section 102(b). *See*
 15 *Samuelson*, 85 TEX. L.R. at 1923 ("To be more consistent with § 102(b), courts would be well

16 ⁵ Google has investigated the availability of the files Oracle accuses of literal copying on
 17 Google's website.

18 As Google indicated at the recent pre-trial conference, the accused portions of the files have all
 19 been deleted from the current release (Ice Cream Sandwich) of Android, starting at version 4.0.

20 The rangeCheck() method is, however, part of versions of Android prior to Ice Cream Sandwich.
 21 Google does not believe it has represented otherwise.

22 Although Oracle's counsel asserted that the "comments" in two files are still in version 4.0.1,
 23 Google believes that claim is in error. These comments have been removed from Android since
 24 version 2.3.3r1 of Android, over a dozen version releases ago.

25 Likewise, the other eight test files were removed from Android since version 2.3.3r1, over a
 26 dozen version releases ago. They were, however, part of version 2.2.3r2, which was a subsequent
 27 security patch to the "2.2 branch" of the Froyo release of Android. As of March 30, 2012, this
 28 has been superseded by version 2.2.3r2.1, in which the eight test files have been deleted..

As explained at the pre-trial conference, due to the way the Android source code repository and
 revision control system operates, it is possible to request old, out-of-date versions of Android
 from the Google website, and the revision control software can "recreate" these old releases,
 which may contain the accused portions of the files.

In any event, Google maintains that any alleged copying in these files is *de minimis* and thus not
 actionable. Moreover, aside from rangeCheck(), none of the alleged literal copying affected any
 code that shipped on any Android phone, because the other files at issue are test files and/or the
 allegedly copied material was only in comments to code, not code itself.

1 advised to speak of the ‘protectable/unprotectable distinction’ in copyright law.”).

2 The selection, arrangement and structure of the APIs present a different situation. At any
 3 level of abstraction above the actual implementation (i.e., the source code), the APIs *no longer*
 4 *meet the statutory definition of a computer program.* This is because the APIs are a general-
 5 purpose collection of classes, interfaces, methods and fields that can be arranged in whatever
 6 order a developer chooses, to bring about whatever particular result that developer desires.
 7 Nothing in the structure, sequence and organization of the APIs dictates whether the max()
 8 method should be run before or after the getContent() method, or any other method. One
 9 developer may use the elements to create a video game, while another developer might use the
 10 elements in a different manner to create a word processor, and yet another developer might
 11 instead create a program for electronic banking.

12 The classes, interfaces, methods and fields in the APIs at issue are merely tools waiting to
 13 be used by a developer who is writing his or her own application. Thus, if viewed at a level of
 14 abstraction above the source code level, the APIs are not “a set of statements or instructions to be
 15 used directly or indirectly in a computer in order *to bring about a certain result.*” 17 U.S.C.
 16 § 101 (emphasis added). In other words, divorced from implementing code, the APIs are not
 17 computer programs. Instead, the APIs are merely the medium through which Java language
 18 developers express themselves. Here, with all due respect to Marshall McLuhan, the medium is
 19 not the message. It is a *system* that can be *used* to express. And a system, by definition, is
 20 outside the realm of copyright protection. *Id.* § 102(b).

21 This view of Section 102(b) is consistent with the legislative history:

22 Some concern has been expressed *lest copyright in computer programs should*
 23 *extend protection to the methodology or processes adopted by the programmer,*
 24 *rather than merely to the “writing” expressing his ideas.* Section 102(b) is
 25 intended, among other things, to make clear that the expression adopted by the
 programmer is the copyrightable element in a computer program, and that the
 actual processes or methods embodied in the program are not within the scope of
 the copyright law.

26 See H.R. Rep. No. 94-1476, at 57 (1976), reprinted in 1976 U.S.C.C.A.N 5659, 5670 (emphasis
 27 added). Section 102(b) is intended “to restate, in the context of the new single Federal system of
 28 copyright, that the basic dichotomy between expression and idea remains unchanged.” *Id.* As

1 Professor Samuelson has correctly noted, “[i]dea,’ as used in this context, should be understood
 2 as shorthand for the eight terms of exclusion set forth in § 102(b).” *Samuelson*, 85 TEX. L.R. at
 3 1952.

4 Ninth Circuit law is not to the contrary. Whether, at a given level of abstraction, the
 5 “structure, sequence and/or organization” of a computer program “is protected by a copyright
 6 depends on whether it qualifies as an ‘expression’ of an idea, rather than the idea itself.” *Johnson*
 7 *Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989). In *Johnson*
 8 *Controls* the plaintiff’s work was software for wastewater treatment plants, and the defendants
 9 wrote a similar program. *See id.* at 1174, 1176. The Ninth Circuit held that the district court had
 10 not clearly erred in finding that the defendants had copied protectable expression by copying the
 11 structure, sequence and organization of the plaintiff’s software. *Id.* at 1175-76. The decision
 12 offers little insight into what structure, sequence and organization were copied, noting only:

13 We conclude that Johnson met its burden sufficiently to support the preliminary
 14 injunction. The special master’s report sets forth, in detailed form, the various
 15 similarities between the programs. These similarities, both in idea and expression,
 16 would permit a reasonable person to find an unlawful appropriation, a capture by
 17 the infringing work of the “total concept and feel” of Johnson’s work.

18 *Id.* at 1176 (footnote and citation omitted). Because this was an appeal of a preliminary
 19 injunction, the court reviewed the district court’s decision only for clear error. *See id.* Similarly,
 20 in finding that the district court had not clearly erred in holding that the structure, sequence and
 21 organization was expression rather than idea, the court noted,

22 This issue will no doubt be revisited at trial, but *at this stage of the proceedings*
 23 we cannot say that the district court clearly erred.

24 *Id.* (emphasis added). Given the limited discussion of the facts and the preliminary procedural
 25 posture of the case, it is impossible to draw from *Johnson Controls* a general rule governing when
 26 a structure, sequence and organization of a computer program (much less the structure, sequence
 27 and organization of interfaces used by a computer program) is copyrightable.⁶ Finally, if the
 28 allegedly copied structure, sequence and organization is a system or method of operation, it is *per*

27 ⁶ Moreover, *Johnson Controls* pre-dates *Feist*, so any application of *Johnson Controls* must keep
 28 in mind that the structure, sequence and organization of a computer program are copyrightable
 only if the structure, sequence and organization are *creative*. *See Feist*, 499 U.S. at 350.

1 se barred from copyright protection. *See* 17 U.S.C. § 102(b).

2 On the present facts, Oracle's claim fails. Programmers cannot use the APIs without
 3 relying on the structure, sequence and organization of the APIs. Whether this means that the
 4 APIs are a *system* for expression, or that they are a *method of operation*, *see Lotus Dev. Corp. v.*
 5 *Borland Int'l, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995), *aff'd by an equally divided court*, 516 U.S.
 6 233 (1996), the APIs are uncopyrightable.⁷ As Oracle's own expert has opined, "An API
 7 specification does not actually run on a computer; rather, it describes *a set of rules* that the code
 8 implementing the library must follow." Mitchell Copyright Report [Dkt. 341-1] ¶ 175 (emphasis
 9 added). As he further has explained:

10 An API consists of *a set of names that can be used to access features of the*
 11 *library, together with specified conventions about their use.* For example, an API
 12 allowing a program to determine the *time* of day might include a function called
 time, together with the convention that a call to this function returns an integer
 representing the clock time in a particular format.

13 *Id.* ¶ 52 (first emphasis added). What Dr. Mitchell has described is a system or a method of
 14 operation.

15 The structure, sequence and organization at issue here embody concepts too inchoate to
 16 qualify as protectable *expression*. The many design choices Oracle relies upon to show
 17 "creativity" all embody *ideas* about how a programming language should be structured. The
 18 accused aspects of the APIs are uncopyrightable. 17 U.S.C. § 102(b).

19 **D. The merger doctrine is not limited to high levels of abstraction, and bars**
 20 **copyright protection for any arguable expression in the selection,**
 arrangement and structure of the APIs.

21 The Court has asked, "[f]or the merger doctrine, at what level of abstraction should we
 22 consider the idea/system?" 3/29/12 Order [Dkt. 838], Question 4. At a high level of abstraction,
 23 such as "the concept of APIs generally," the Court suggests that there are many different ways to
 24 select, arrange and structure the APIs. *Id.* At the level of the idea/system of the 37 Java APIs,
 25 however, "there may be only one way to express the 'selection, arrangement and structure.'" *Id.*
 26 The merger doctrine is not limited to high levels of abstraction.

27 ⁷ There also is no evidence that the particular structure, sequence and organization of the 37 API
 28 subset of J2SE 5.0's 166 APIs is creative.

1 For example, in *Allen v. Academic Games League of America, Inc.*, 89 F.3d 614 (9th Cir.
 2 1996), the plaintiff Allen owned the copyright to several games, including to the manuals that he
 3 wrote and that accompanied those games. *Id.* at 615-16. The defendant game league
 4 (“AGLOA”) organized tournaments for the playing of these games, and created tournament
 5 rulebooks. *Id.* at 616. AGLOA’s tournament rulebooks included some of the rules described in
 6 Allen’s game manuals. *Id.* at 617. The Ninth Circuit noted that “Allen has not shown that it is
 7 possible to distinguish the expression of the rules of his game manuals from the idea of the rules
 8 themselves.” *Id.* at 618. The court held that the merger doctrine barred Allen’s claim of
 9 infringement:

10 Thus, the doctrine of merger applies and although Allen may be entitled to
 11 copyright protection for the physical form of his games, he is not afforded
 12 protection for the premises or ideas underlying those games. To hold otherwise
 would give Allen a monopoly on such commonplace ideas as a simple rule on how
 youngsters should play their games.

13 *Id.* The Ninth Circuit did not limit the doctrine of merger to deciding whether there is only one
 14 way or a limited number of ways to play games. The Ninth Circuit did not even limit its inquiry
 15 to the number of ways to play academic games covering the subject matter of Allen’s games.
 16 Plainly, had the Ninth Circuit taken either of those approaches, the court would have concluded
 17 that AGLOA could have designed its own games, with its own rules, that addressed the same
 18 high-level concepts as Allen’s games. Instead, the Ninth Circuit held that the merger doctrine
 19 prevented Allen from asserting copyright protection over his expression of *his particular rules* for
 20 *his particular games*. *See id.*

21 In its most recent brief, Oracle cited *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335 (5th Cir. 1994) (“*EDI I*”), for the proposition that where input/output formats
 22 could have been structured in “numerous ways,” an original “selection, sequence and
 23 coordination of inputs” can be copyrightable. Oracle’s 3/27/12 Br. [Dkt. 833] at 2 (quoting *EDI I*, 26 F.3d at 1344-46). Oracle neglected, however, to discuss the subsequent order in which the
 24 Fifth Circuit limited the reach of its *EDI I* holding. *See* 46 F.3d 408 (5th Cir. 1995) (“*EDI II*”)
 25 (order denying petition for rehearing *en banc*). In this “supplement to the panel opinion to avoid
 26 any confusion as to its scope,” the court explained:
 27
 28

1 1. The petition for rehearing and amicus petitions in support of rehearing suggest
 2 that our initial opinion held that EDI's user formats are not only protectable but
 3 protected by copyright law because there "are numerous ways the input formats
 4 could be organized." ***This is an overly simplistic view of the opinion.*** The panel
 5 adheres to its adoption of the abstraction-filtration-comparison test and to its
 6 application of the test on the facts before us. *See e.g., Computer Assoc's Int'l Inc.*
v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992). One of those facts, among many, was
 7 that EDI created input/output formats to facilitate the engineering and design of
 8 offshore structures and in this context, the formats were more than a "blank form."
The panel did not say that in any case involving user interface the fact that the
"author" has selected from among possible formats is dispositive.

9 *Id.* at 409 (emphases added). To the contrary, the court held that "copyright only protects
 10 originality of user interface *to the extent* that the selection of variable inputs from the universe of
 11 potential inputs reflects non-functional judgments." *Id.* (citing *Gates Rubber Co.*, 9 F.3d at 838).

12 The court further cautioned that:

13 2. ***This opinion cannot properly be read*** to extend to the manufacturing of
 14 computer hardware so as ***to deter achieving compatibility*** with other models or to
 15 the practice employed by users of programs of analyzing application programs to
 16 "read" the file formats of other programs.

17 *EDI II*, 46 F.3d at 410 (emphases added) (citing, among other cases, *Sega Enters. Ltd. v.*
 18 *Accolade, Inc.*, 977 F.2d 1510, 1525-27 (9th Cir. 1992)).

19 Here, the selection, structure and arrangement of the 37 APIs reflect judgments about
 20 what APIs are most useful, what options should be made available to developers, and how best to
 21 organize the APIs. These choices were precisely the types of "functional judgments" the Fifth
 22 Circuit explained *cannot* be protected by copyright. *See EDI II*, 46 F.3d at 409. Oracle cannot
 23 seriously claim otherwise. Each of the APIs individually, and all of them collectively, were
 24 selected and coordinated to enhance the capabilities and usability of J2SE, not for any aesthetic or
 25 stylistic purpose. Moreover, these choices dictate precisely the functional requirements for
 26 compatibility that the Ninth Circuit has held cannot be copyrighted in light of Section 102(b). *See*
Sega, 977 F.2d at 1522. Thus, even if the Court were to conclude that some aspects of the
 27 selection, arrangement and structure of the APIs are expressive, there is no way to "express" the
 28 idea of the 37 APIs without repeating that selection, arrangement and structure. The merger
 29 doctrine therefore bars copyright protection for any arguable expression in the selection,
 30 arrangement and structure of the APIs.

1 **E. The Java language APIs at issue are deeply intertwined with and integral to**
 2 **the Java programming language.**

3 It is undisputed that the Java programming language is not protected by copyright. *See*
 4 Google Reply Trial Br. [Dkt. 823] at 4-5. The Court has asked whether the APIs are an integral
 5 part of Java—whether “programmers write their own programs using the APIs.” 3/29/12 Order
 6 [Dkt. 838], Question 5. They are integral, and programmers do exactly that.

7 Sun’s own specification for the Java *language* repeatedly references the APIs. *See The*
 8 *Java Language Specification, Third Edition*, Trial Ex. 984 at 6 (“Throughout this book we refer
 9 to classes and interfaces drawn from the Java and Java 2 platforms.”). For example, when
 10 discussing variable types, the language specification notes that “useful constructors, methods, and
 11 constants are predefined in the classes Byte, Short, Integer, Long, and Character,” *id.* at 36, all of
 12 which are classes defined in the java.lang API package.

13 The language specification further notes that “some classes” from the APIs “have a
 14 special relationship with the Java programming language,” including “Object, Class,
 15 ClassLoader, String, Thread, and the classes and interfaces in package java.lang.reflect, among
 16 others.” *Id.* at 6. These classes are *required* to behave in certain ways to conform to the
 17 *language* specification. *See id.*

18 In short, there is no way to implement the Java programming language without
 19 implementing at least some of the accused APIs. As a purely technical matter, there is no bright
 20 line distinction between the Java programming *language* and the Java language *APIs*. Beyond
 21 this simple technical fact, however, *all* of the APIs at issue either are fundamental to the
 22 language, or so commonly used by developers that they should be treated as part of the language
 23 for purposes of Oracle’s copyright claims.

24 **1. More than half of the accused API packages are described by Sun as**
 25 **“fundamental” to the Java programming language.**

26 Sun’s 1996 two-volume series describing one of the first releases of the Java language
 27 APIs call the APIs “a standard set of libraries for writing Java programs.” Trial Ex. 980 at xvii.
 28 Sun explained that it believed the APIs were “useful and [that it] hope[d] to make [them] a
 29 *ubiquitous layer, available to all Internet applications.*” *Id.* (emphasis added). These books were

part of Sun’s “Java Series,” described by Sun as “the definitive reference documentation for Java programmers and end users.” *Id.* at xvi. The books do not purport to be a reference only for those who specifically want to use the APIs; they are “reference manuals for Java application and applet programmers,” without qualification. *Id.* at xviii. “The API[s] documented in this book will remain available to *all Java programs* through future releases.” *Id.* (emphasis added).

In this early release, there were only eight API packages. Sun described *four* of them (java.lang,⁸ java.io, java.util and java.net) as “general-purpose libraries *fundamental to every Java program.*” *Id.* (back cover) (emphasis added). As the libraries have grown, those four “fundamental” libraries have expanded to include subpackages, including three accused “lang” subpackages (java.lang.annotation, java.lang.ref and java.lang.reflect⁹), five accused “util” subpackages (java.util.jar, java.util.logging, java.prefs, java.util.regex and java.util.zip), and two extensions to the “net” packages (javax.net and javax.net.ssl). Five of the other accused packages are the “new input/output” (“nio”) package and subpackages (java.nio, java.nio.channels, java.nio.channels.spi, java.nio.charset, and java.nio.charset.spi), which are closely related to the “io” packages. Collectively, these packages that even Sun described as “fundamental” to the language account for 19 of the 37 accused packages.

2. The other 18 accused API packages relate to features that are standard in modern applications.

Three additional accused packages relate to programming vocabulary that is commonplace in the Java language programming community. First, the java.awt.font package includes basic APIs for fonts (i.e., typefaces). Second, the java.beans package includes APIs necessary for developing components known as “beans,” which have been commonly used by Java language programmers. Third, the java.text package includes APIs that allow applications to be “localized” more easily—that is, adjusting things like the date and time format, or the (human) language used, based on location.

⁸ The java.lang package in particular is described by Oracle’s specification as “[p]rovid[ing] classes that are fundamental to the design of the Java programming language.” Trial Ex. 3425.

⁹ As already noted, the java.lang.reflect package is *required* to be implemented by the language specification. See Trial Ex. 984 at 6.

1 Moreover, many modern applications implement features requiring security (such as
 2 secure communications), cryptography (on which the security packages depend) and databases
 3 (such as storing data generally). These features are common for smartphone applications as well.
 4 The remaining 15 accused Android API packages implement the API specifications in these areas
 5 that Java language programmers are likely to expect to be available for a smartphone platform.
 6 This includes ten security API packages and subpackages (java.security, java.security.acl,
 7 java.security.cert, java.security.interfaces, java.security.spec, javax.security.auth,
 8 javax.security.auth.callback, javax.security.auth.login, javax.security.auth.x500, and
 9 javax.security.cert), three cryptography API packages and subpackages (javax.crypto,
 10 javax.crypto.interfaces, and javax.crypto.spec), and two database API packages (java.sql and
 11 javax.sql).

12 **3. The 37 Java language API packages are all integral to the Java
 13 programming language.**

14 Oracle cannot seriously dispute that Java language programmers *treat* the APIs as part of
 15 the language. Indeed, all 37 of the disputed APIs have long been a part of both GNU Classpath
 16 and Apache Harmony, open-source implementations offered by the Free Software Foundation
 17 and the Apache Foundation to the Java language programming community. Neither of these
 18 implementations is licensed by Oracle, and yet Oracle has taken no action to prevent others from
 19 using either of them.

20 Instead of disputing that Java language programmers treat the APIs as part of the
 21 language, Oracle argues that *technically* one could write programs in the Java language without
 22 using the APIs (or at least without using some of the APIs). This is like saying one can write a
 23 novel in English without the letter ‘e.’ *See* Ernest Vincent Wright, *Gadsby: A Story of Over*
 24 *50,000 Words Without Using the Letter “E”* (Wetzel Publ. Co. 1939). It technically could be
 25 done, but that doesn’t mean the letter ‘e’ isn’t integral to the English language. *Cf.* Guy L. Steele,
 26 *Growing a Language* (Sun Microsystems, Oct. 1998) (using no multisyllabic words without first
 27 defining them). By seeking copyright protection over common, everyday vocabulary used by
 28 millions of Java language programmers, Oracle ignores the common import of its concession that

1 it does not assert copyright over the Java programming language.

2 **F. Computer languages are neither copyrightable nor patentable.**

3 The Court has asked “[t]o what extent are computer languages (not programs, but
4 languages) copyrightable? Patentable?” 3/29/12 Order [Dkt. 838], Question 6. Computer
5 languages are neither copyrightable nor patentable, for the reasons given below.

6 **1. Computer languages are not copyrightable.**

7 As explained above, the APIs are an uncopyrightable system or method of operation. A
8 computer program language is equally uncopyrightable, for the same reasons. *See supra*, Part
9 II.C; 17 U.S.C. § 102(b). While there is no case under the Copyright Act that is directly on point,
10 the High Court of Justice of England and Wales recently concluded that computer programming
11 languages are not copyrightable. *SAS Institute, Inc. v. World Programming Ltd.*, [2010] EWHC
12 (Ch) 1829 (Eng.).¹⁰

13 Although this decision applied English law rather than the Copyright Act, the underlying
14 principles are identical. The High Court began its analysis by considering several international
15 intellectual property treaties, including the Agreement on Trade-Related Aspects of Intellectual
16 Property Rights (“TRIPs”) and the World Intellectual Property Organization Copyright Treaty
17 (“WIPO Copyright Treaty”). *Id.* ¶¶ 150-152.¹¹ The High Court recognized that both the TRIPs
18 agreement and the WIPO Copyright Treaty state that copyright protection extends to expressions,
19 but not to “ideas, procedures, methods of operation or mathematical concepts as such.” *See id.*
20 ¶¶ 151-152 (quoting TRIPs, Article 9 and WIPO Copyright Treaty, Article 2). It also noted that
21 these two treaties specify that both computer programs and compilations of data or other material
22 can be covered by copyright. *Id.* (quoting TRIPs Article 10, WIPO Copyright Treaty Articles 4,
23 5). The High Court devoted significant attention to discussing the distinction between ideas and
24 expression, mainly discussing EU and English law, but also citing 17 U.S.C. § 102(b). *See id.*
25 ¶¶ 199-207.

26 ¹⁰ Available at <http://www.bailii.org/ew/cases/EWHC/Ch/2010/1829.html>.

27 ¹¹ The United States is also a signatory to both the TRIPs agreement and the WIPO Copyright
28 Treaty.

The plaintiff in the case, SAS, produces analytical software known as the SAS System that enables users to carry out a wide range of data processing and statistical analysis tasks. *Id.* ¶ 1. The core component of the SAS System is Base SAS. Base SAS allows users to run application programs written in the SAS Language. *Id.* The SAS language consists primarily of DATA steps and PROC steps. *Id.* ¶ 36. DATA steps are the primary methods for creating or manipulating data. *Id.* A PROC is a software tool written by SAS Institute to perform a wide variety of particular types of data analysis and reporting, such as producing statistics, tables, reports, and charts. *Id.* ¶ 38. For example, two PROC sets are called LOGISTIC and UNIVARIATE. *Id.* ¶ 39. In this sense the PROCs are similar to the APIs at issue in this case.

World Programming recognized that there might be a market for alternative platforms that could read applications written in the SAS language. *Id.* ¶ 3. It therefore developed the World Programming System, or WPS. With some exceptions, WPS was written so that it could execute programs written in the SAS language. *Id.* ¶¶ 3, 68, 73. In developing WPS, World Programming did not copy any of the SAS source code. *Id.* ¶¶ 3, 69. Thus just as Google followed the selection, arrangement and structure of 37 of the Java language API specifications in Android but wrote its own source code, World Programming used the PROC names and structure from SAS, but wrote its own source code. SAS brought a series of copyright claims against World Programming alleging, among other things, that World Programming violated its copyrights by implementing the SAS Language. *Id.* ¶ 211.

The High Court concluded that the SAS Language constitutes an idea, not expression, and thus is not subject to copyright protection. Referencing a decision in a prior case, the High Court wrote:

I think that the distinction which Pumfrey J drew between a computer program and the language it is written in is, despite his hesitancy on the point, perfectly consistent with the distinction between expressions and ideas, procedures, methods of operation and mathematical formulae.

Id. ¶ 217. Thus the High Court concluded that World Programming's reproduction of the SAS Language did not violate SAS's copyrights. *Id.* ¶ 247.

The High Court also referred the question to the European Court of Justice. *Id.* ¶ 227.

1 While the ECJ has not yet ruled, an ECJ Advocate General recently agreed with the High Court.¹²
 2 Advocate General Bot concluded that under EU law, a computer programming language is not
 3 protected by copyright. Advocate General's Opinion, *SAS Institute Inc. v. World Programming*
 4 *Ltd.*, Case C-406/10 (Nov. 29, 2011), ¶ 75.¹³ Also focusing on the distinction between ideas and
 5 expression, the Advocate General concluded:

6 It seems to me, therefore, that programming language is a functional element
 7 which allows instructions to be given to the computer. As we have seen with SAS
 8 language, programming language is made up of words and characters known to
 9 everyone and lacking any originality. In my opinion, programming language must
 be regarded as comparable to the language used by the author of a novel. ***It is
 therefore the means which permits expression to be given, not the expression
 itself.***

10 *Id.* ¶ 71 (emphasis added).

11 The decision from the English High Court and the recommendation from the ECJ
 12 Advocate General fully support Google's position. These opinions, while not binding precedent,
 13 analyze a set of facts and legal principles that for all intents and purposes are identical to those
 14 presented in the case at bar. *See also* Richard H. Stern, *Copyright in Computer Programming*
 15 *Languages*, 17 RUTGERS COMPUTER & TECH L.J. 321 (1991) (concluding that computer
 16 programming languages are not copyrightable).

17 2. Computer languages are not patentable.

18 An entire programming language also is not patentable. The Patent Act "specifies four
 19 independent categories of inventions or discoveries that are eligible for protection: processes,
 20 machines, manufactures, and compositions of matter." *Bilski v. Kappos*, 130 S. Ct. 3218, 3225
 21 (2010) (discussing 35 U.S.C. § 101). A programming language is none of these things.

22 A programming language is not a process. Although the set of steps constituting a
 23 program written in that language could arguably be a process, the language itself, from which the
 24 steps would be constructed, is not itself a process. It could at most be a tool by which to create a
 25 process.¹⁴

26 ¹² The Advocate General is similar to a special master before the Supreme Court in that he hears
 27 argument from the parties and gives a legal opinion to the ECJ.

28 ¹³ Available at <http://curia.europa.eu/juris/celex.jsf?celex=62010CC0406&lang1=en&type=NOT>.

14 Moreover, for purposes of patent law, even a program written in a programming language may
 16

1 A programming language is not a machine. Although a program written in a
 2 programming language and installed on a general purpose computer could create a patentable
 3 machine,¹⁵ this is only applicable to a specific program that is installed on a given machine, not
 4 an entire programming language, which by itself is not designed to perform any particular
 5 functions.

6 Finally, a programming language is neither a manufacture nor a composition of matter. *In*
 7 *re Ferguson*, 558 F.3d 1359, 1366 (Fed. Cir. 2009) (manufacture is an ““article’ resulting from
 8 the process of manufacture.” (quoting *In re Nuijten*, 500 F.3d 1346, 1356 (Fed. Cir. 2007)); *In re*
 9 *Comiskey*, 554 F.3d 967, 977 n.10 (Fed. Cir. 2009) (quoting *Diamond v. Chakrabarty*, 447 U.S.
 10 303, 308 (1980)) (composition of matter includes “all compositions of two or more substances
 11 and all composite articles, whether they be results of chemical union, or of mechanical mixture,
 12 or whether they be gases, fluids, powders or solids.”).

13 Given that a programming language fails even the threshold test for patent eligibility
 14 under Section 101, it cannot be patentable. *See also* Sebastian Zimmeck, *Patent Eligibility of*
 15 *Programming Languages and Tools*, 13 TUL. J. TECH. & INTELL. PROP. 133, 152 (2010) (“As
 16 grammars are the defining elements of programming languages, it follows that programming
 17 languages are not patent-eligible.”). Notably, the reason a computer programming language fails
 18 to qualify as patentable subject matter is, in essence, that it lacks the type of determinate steps that
 19 are a threshold requirement to qualify it as a process or a machine implementing a particular
 20 process. The fact that a programming language is not patentable subject matter thus further
 21 supports the conclusion that it is on the “idea” side of the idea/expression dichotomy.

22
 23
 24 not be a process: “Since a computer program is merely a set of instructions capable of being
 25 executed by a computer, the computer program itself is not a process and USPTO personnel
 26 should treat a claim for a computer program, without the computer-readable medium needed to
 27 realize the computer program’s functionality, as nonstatutory functional descriptive material.”
 28 Manual of Patent Examining Procedure § 2106.01.

¹⁵ *In re Alappat*, 33 F.3d 1526, 1545 (Fed. Cir. 1994) (en banc) (“such programming creates a
 new machine, because a general purpose computer in effect becomes a special purpose computer
 once it is programmed to perform particular functions pursuant to instructions from program
 software”).

1 **G. Subject to the requirements of the Patent Act, it may be possible to try to
2 claim the selection of classes for APIs under patent law.**

3 The Court has asked whether it would “be possible to claim the selection of classes for
4 APIs under patent law.” 3/29/12 Order [Dkt. 838], Question 7. As the Court notes, Sun and
5 Oracle have tried to claim the structure, hierarchy and arrangement of certain APIs in claim 1 of
6 U.S. Patent No. 6,598,093 B1 (the “’093 patent”).

7 It may be possible to try to claim the selection of classes for APIs as part of a patent-
8 eligible claim. Presuming that the claim as a whole satisfies 35 U.S.C. § 101 (and the other
9 requirements of the Patent Act), if it relates to a system, method, or computer-readable medium
10 that includes one or more claim elements directed to the selection of classes for APIs, those
11 elements may be protected by patent law. Sun appears to claim a selection of classes for APIs in
12 claims 4, 5 and 6 of the ’093 patent. In the ’093 patent, independent claim 1 relates to a class
13 structure in a computer system, with three sets of classes arranged in three API sets. Dependent
14 claims 4, 5, and 6 claim, then list, particular classes for inclusion in each API set, respectively.

15 4. A class structure as recited in claim 3 wherein the *at least one abstract
16 class includes a BaseEntry class, and the at least one concrete class includes a
17 SystemDatabase class, a Query class, and a PropertyQuery class.*

18 5. A class structure as recited in claim 4 wherein *the second set of classes
19 includes a SystemEntry class*, the SystemEntry class being a concrete class.

20 6. A class structure as recited in claim 5 wherein *the third set of classes
21 includes a SystemAliasEntry class and a PersistentSystemEntry class*, wherein
22 the SystemAliasEntry class and the PersistentSystemEntry class are concrete
23 classes.

24 ’093 Patent at col. 12, ll. 19-30 (emphases added).

25 These claims are, of course, subject to challenge in the same manner as are all patent
26 claims. And the rules applicable to patents are, in many respects, more stringent than those
27 applicable to copyright. *Compare, e.g., 35 U.S.C. §§ 102-103 with Baker v. Selden*, 101 U.S. 99,
28 102 (1879) (“The copyright of the book, if not pirated from other works, would be valid without
regard to the novelty, or want of novelty, of its subject-matter.”).

29 That is, however, precisely the point. Allowing copyright protection for a system such as
30 the selection, arrangement and structure of the APIs at issue would allow “exclusive property in

1 the art described therein, when no examination of its novelty has ever been officially made,”
 2 which “would be a surprise and a fraud upon the public.” *Baker*, 101 U.S. at 102. “That is the
 3 province of letters-patent, not of copyright. The claim to an invention or discovery of an art or
 4 manufacture must be subjected to the examination of the Patent Office before an exclusive right
 5 therein can be obtained; and it can only be secured by a patent from the government.” *Id.*

6 **H. Aside from its “selection, arrangement and structure” argument, Oracle’s
 7 only argument that Google’s *implementation* of the APIs infringes is based on
 8 nine lines of mundane code from Arrays.java.**

9 The Court asks, with respect to Google’s implementation of the APIs, whether “Oracle is
 10 alleging that Google copied something other than the ‘selection, arrangement, and structure’ of
 APIs, as fixed in the specification.” 3/29/12 Order [Dkt. 878], Question 8.

11 Aside from the nine-line rangeCheck() method in Arrays.java, the answer is “no.” At the
 12 recent pre-trial conference, the Court asked:

13 Except for rangeCheck, the APIs that you have, the 37 APIs in Android compared
 14 to the 37 analogs in Java have different source code. True?

15 3/28/12 Tr. at 38-39. This was quickly followed up with the following exchange:

16 THE COURT: All right. So once you get past the declarations, is your source
 code different than the Java source code?

17 MR. KWUN: Yes.

18 THE COURT: Is that part agreed to?

19 MR. JACOBS: *Yes, Your Honor.*

20 *Id.* at 41 (emphasis added).

21 Moreover, there is no dispute that the rangeCheck() method is merely nine lines out of
 22 over three thousand lines of code in Oracle’s Arrays.java file.¹⁶ Qualitatively, the method is
 23 trivial. *See id.* at 21 (Court’s statement that “even I know enough to know that the concept of
 24 rangeCheck is—that’s like the first grade, isn’t it?”).

25
 26
 27

¹⁶ The nine lines are an equally small part of the Android files in which they appear. Measured
 28 against the work as a whole, whether that work is J2SE 5.0 or the Android platform, the nine lines
 are almost vanishingly small.

III. CONCLUSION

For the foregoing reasons, Google respectfully requests that the Court hold that the selection, arrangement and structure of the APIs are uncopyrightable.

Dated: April 3, 2012

KEKER & VAN NEST LLP

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

Attorneys for Defendant
GOOGLE INC.