

Catholic University Journal of Law and Technology

Volume 24 | Issue 1


Article 10

December 2015

Oracle America, Inc. v. Google, Inc., 750 F.3d 1339 (Fed. Cir. 2014), Cert. Denied: Ideas, Methods, And Expression - Whose Innovation is Protected?

Christopher J. Geissler
Catholic University of America, Columbus School of Law

Follow this and additional works at: <https://scholarship.law.edu/jlt>

 Part of the [Commercial Law Commons](#), [Communications Law Commons](#), [Computer Law Commons](#),
[Intellectual Property Law Commons](#), and the [Science and Technology Law Commons](#)

Recommended Citation

Christopher J. Geissler, *Oracle America, Inc. v. Google, Inc.*, 750 F.3d 1339 (Fed. Cir. 2014), *Cert. Denied: Ideas, Methods, And Expression - Whose Innovation is Protected?*, 24 Cath. U. J. L. & Tech (2015).
Available at: <https://scholarship.law.edu/jlt/vol24/iss1/10>

This Case Summary is brought to you for free and open access by CUA Law Scholarship Repository. It has been accepted for inclusion in Catholic University Journal of Law and Technology by an authorized editor of CUA Law Scholarship Repository. For more information, please contact edinger@law.edu.

***ORACLE AMERICA, INC. V. GOOGLE INC.,
750 F.3D 1339 (FED. CIR. 2014), CERT.
DENIED: IDEAS, METHODS, AND
EXPRESSION - WHOSE INNOVATION IS
PROTECTED?***

Christopher J. Geissler*

The Federal Circuit recently expanded the protections of the Copyright Act to thirty-seven packages of Oracle’s application program interface because Java’s declaring source code contains expression and is not exclusively functional. What does this mean and what are its implications?

I. THE JAVA PACKAGES

Java created 166 application program interfaces (known as “APIs” or “packages”) which function as shortcuts for programmers to build certain functions into their programs.¹ APIs are split into declaring code and implementing code.² Declaring code is the expression that identifies the pre-written function.³

* J.D. Candidate, The Catholic University of America: Columbus School of Law, 2016; B.A. Political Communication & Political Science, The George Washington University, 2011.

¹ Brief of The Open Source Initiative at 9, Mozilla Corp., and Engine as Amicus Curiae, *Google Inc., v. Oracle Am., Inc.*, 750 F.3d 1339 (Fed.Cir. 2014).

Similarly [to computer programs], interoperability dictates the standardization of light bulb bases. Because light bulb bases are standardized, a purchaser has access to a multitude of cheap options when selecting a bulb. No matter the bulb, the purchaser can be confident that it will work with any given base to transfer electricity from the socket in the lamp to the light bulb. A homeowner could have a black-light bulb, a fluorescent bulb, or an energy-efficient bulb, yet because the socket is standardized, none of those outputs matter to the electrician who wires a lamp. In the same way, an API allows two programs to interact with each other through a standardized language.

Id.

² Kacee Taylor, Note, *Oracle Am., Inc. v. Google Inc. 750 F.3d 1339 (Fed Cir. 2014)*, 19 INTELL. PROP. L. BULL. 221 (2015).

³ Petition for Writ of Certiorari, at 5, *Google, Inc. v. Oracle Am., Inc.*, 135 S.Ct. 2887

Implementing code provides instructions for carrying out the functions.⁴ Google did not dispute that it copied 37 of those API packages in its formation of the Android operating system for mobile devices but it wrote its own implementing code.⁵ The parties' dispute centered on Google's use of the same headers for the methods found in 37 of the Android packages—methods that perform necessary functions for mobile devices.⁶

Oracle purchased Java in 2010 and offers Java to the public through three different licenses: 1) a free "open source" license which requires that the user "contribute back" its innovations to the public; 2) a Specification License which provides the licensee can use declaring code but must write implementing code; and 3) a Commercial license for businesses who want to use and customize Java code while keeping code secret.⁷ The Specification and Commercial Licenses require that the licensees' programs remain compatible with Java. Oracle argues that the API packages' structure, sequence, and organization ("SSO") is copyrightable, Google used Java's declaring code, and therefore Android is no longer compatible with Java in violation of the open source license.⁸

In 2012, Oracle brought suit in the Northern District of California against Google for its copyright claims.⁹ In finding that the Java APIs were not copyrightable because they were a command structure, system, or method of operation, the district held that Google "remains free to write its own code to carry out the identical function so long as the implementing code in the method body is different from the copyrighted implementation."¹⁰ In 2014, the United States

(2015) (No. 14-410).

⁴ Taylor, *supra* note 2, at 221.

⁵ Oracle Am., Inc. v. Google, Inc. (*Google I*), 872 F.Supp.2d 974, 975 (N.D. Cal. 2012).

⁶ Petition for Writ of Certiorari, *supra* note 3, at 7.

⁷ Opening Brief and Addendum of Plaintiff-Appellant at 14, Oracle Am., Inc. v. Google, Inc., 872 F.Supp.2d 974 (N.D. Cal. 2014).

⁸ *See id.* at 3-5.

⁹ *See Google I*, 872 F.Supp.2d at 974. The parties agreed that the district court would decide copyrightability, so the jury was instructed to assume that the SSO of the APIs was copyrightable, that the 37 packages had been copied verbatim, and that Google coded copying the rangeCheck function and eight decompiled security files, but that Google maintained the violation was de minimis. The jury found infringement in all but the eight security files and the jury hung on Google's fair use defense. The district court granted Oracle judgment as a matter of law of infringement on the eight security files, but found that the declaring code within the SSO of the Java API's was not copyrightable under the merger doctrine forbidding copyright for the only way something can be done. Essentially, the district court found that the SSO resembled a taxonomy, which would be expressive, but instead found that the API's more accurately resembled a command structure, a system, or method of operation not entitled to copyright. *Id.*

¹⁰ *Id.* at 997.

Court of Appeals for the Federal Circuit held that Oracle's declaring code and the structure, sequence and organization of the thirty-seven Java API packages were expressive and therefore entitled to copyright protection and remanded the issue of the affirmative defense of fair use back to the district court. Google petitioned for certiorari, arguing that the Federal Circuit's holding would "obstruct an enormous amount of innovation in fast-moving, high-technology industries."¹¹ The United States Supreme Court denied certiorari on June 29, 2015.¹²

II. HISTORY OF THE COPYRIGHT LAW OF COMPUTER PROGRAMS

In 1879, the Supreme Court held that a copyrighted book on a novel system of bookkeeping was not infringed by a similar book using a similar plan, which achieved similar results.¹³ The alleged infringer expressed the idea differently through a different arrangement of the columns and used different headings.¹⁴ The Court identified a difference between the idea (the system of bookkeeping) and the expression of the idea (the presentation in the book).¹⁵ In 1954, the Court re-affirmed the principle that a copyright cannot be extended to the expression of an idea and not the idea itself.¹⁶

The Copyright Act of 1976 codified this principle.¹⁷ Section 102(b) broadened it to exclude not only ideas but also procedures, processes, systems, methods of operation, concepts, principles, and discoveries regardless of the form in which it is described, explained, illustrated, or embodied in such work.¹⁸ In 1991 and again in 2003, the Supreme Court held that every idea, theory, and fact in a copyrighted work may be copied at will at the moment of publication.¹⁹ Since their inception, courts have disagreed on the classification

¹¹ Petition for Writ of Certiorari, *supra* note 3, at 1, 3, 5.

¹² *Google, Inc. v. Oracle Am. (Google II)*, 750 F.3d 1339 (Fed. Cir. 2014), *cert. denied*, 135 S.Ct. 2887 (U.S. Jun. 29, 2015) (No. 14-410).

¹³ *See generally* *Baker v. Selden*, 101 U.S. 99, 107 (1879).

¹⁴ *Id.* at 100.

¹⁵ *Id.* at 100-01.

If the complainant's testator had the exclusive right to the use of the system explained in his book, it would be difficult to contend that the defendant does not infringe it, notwithstanding the difference in his form of arrangement; but if it be assumed that the system is open to public use, it seems to be equally difficult to contend that the books made and sold by the defendant are a violation of the copyright of the complainant's book considered merely as a book explanatory of the system. Where the truths of a science or the methods of an art are the common property of the whole world, any author has the right to express the one, or explain and use the other, in his own way.

Id.

¹⁶ *Mazer v. Stein*, 347 U.S. 201, 217 (1954) (differentiating copyright from patent).

¹⁷ Copyright Act of 1976, Pub. L. No. 94-553, 90 Stat. 2541.

¹⁸ 17 U.S.C. § 102(b) (2012).

¹⁹ *Feist Pub., Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 350 (1991); *Eldred v. Ashcroft*,

of computer programs and code within the idea/expression dichotomy.²⁰

The 1980 Amendments defined 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.”²¹ The 1980 Amendments did not enumerate which parts of computer programs, if any, were covered by the Copyright Act.

Whether a non-literal element can be protected depends on a fact-specific analysis of the idea/expression dichotomy.²² Separately, the copying of a program can be literal (verbatim) or non-literal (paraphrased).²³ Non-literal copying requires additional analysis whether the similarities are the result of an application of the same idea or of copying.²⁴

The Third Circuit, in 1983, defined both object code and source code as literary works²⁵ and therefore afforded copyright protection to both.²⁶ While conceding that a *method* which instructs the computer to perform would potentially be protected only by patent law, the Third Circuit identified the programmer’s desire to copyright the instructions themselves, not the method.²⁷

Courts outside the Third Circuit consistently rejected the dicta in *Franklin*²⁸ and holding in *Whelan*²⁹ and found instead interface specifications are uncopyrightable ideas in the idea/expression dichotomy. The U.S. Supreme Court, in *Feist*³⁰, rejected *Whelan* and *Franklin*. The Second Circuit in *Altai*³¹, citing *Feist*, found copyright did not extend to program elements necessary for compatibility. Within ten years of *Feist*, “most courts of appeal that ha[d] subse-

537 U.S. 186, 219 (2003).

²⁰ See e.g., *Tandy Corp. v. Pers. Micro Comps., Inc.*, 524 F.Supp. 171, 171 (N.D. Cal. 1981) (finding a computer program to be a “work of authorship” and a silicon chip as a “medium of expression”), *disapproved of* by *Apple Comp., Inc. v. Franklin Comp. Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983).

²¹ Copyright Amendment Act of 1980, Pub. L. No. 96–517, 94 Stat. 3015.

²² *Johnson Controls, Inc. v. Phoenix Control Sys., Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989) (citing *Harper & Row Publishers, Inc. v. Nation Enters.*, 471 U.S. 539, 547 (1985)).

²³ *Lotus Dev. Corp. v. Borland Int’l*, 49 F.3d 807, 814 (1st Cir. 1995) *aff’d*, 516 U.S. 233 (1996).

²⁴ *Borland*, 49 F.3d at 814 (citing the “Altai test” from *Computer Assoc. Int’l, Inc. v. Altai, Inc.*, 982 F.2d 693, 701 (2d Cir. 1992)).

²⁵ *Franklin*, 714 F.2d at 1249 (“[The] category of ‘literary works’, one of the seven copyrightable categories, is not confined to literature in the nature of Hemingway’s *For Whom the Bell Tolls*. The definition of ‘literary works’ in section 101 includes expression not only in words but also ‘numbers, or other ... numerical symbols or indicia.’”) (citing 17 U.S.C. § 101). The Federal Circuit agreed in 1992. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 838 (Fed.Cir. 1992).

²⁶ *Franklin*, 714 F.2d at 1249.

²⁷ *Id.* at 1251.

²⁸ *Id.*

²⁹ *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1248 (3d Cir. 1986).

³⁰ *Feist*, 499 U.S. at 350.

³¹ *Computer Assocs. Int’l v. Altai, Inc.*, 982 F.2d 693, 711 (2d Cir. 1992).

quently addressed the question of scope of copyright protection for computer programs...effectively rejected the *Whelan* approach.³²

Circuits have differed in their holdings because they have each applied different tests to Section 102.³³ The Second Circuit's "Altai test" created an abstraction-filtration-comparison test that has found favor with other circuits.³⁴ However, the First Circuit, in *Lotus*, was concerned that the abstraction-filtration-comparison approach would create a base level of infringement for any verbatim copying.³⁵

The SSO analysis is another standard for separating the ideas and the expression in computer programs. Both the literal and non-literal elements of a computer program can be protected.³⁶ Source code³⁷ and object code³⁸ are the literal elements of a computer program.³⁹ The program's SSO is a non-literal component.⁴⁰ However, even the name of structure, sequence, and organization might be interpreted to fall within the process, system, or method of operation exemptions in 102(b).

The U.S. Supreme Court held that menu hierarchies that control functional capabilities are a method of operation.⁴¹ The Ninth Circuit specifically found that a "system interface procedure" as part of Sony's Playstation was not copyrightable⁴² within the broader holding that functional requirements for compat-

³² 1 GOLDSTEIN ON COPYRIGHT § 2.15.1 (2d ed. 2005) (citing *Sega Enters. v. Accolade, Inc.*, 977 F.2d 1510, 1521 (9th Cir. 1992)); *Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1371 (10th Cir. 1997); *Atari*, 975 F.2d at 838-39; *see also* *Lexmark Int'l v. Static Control Components, Inc.*, 387 F.3d 522, 534 (6th Cir. 2004).

³³ *Compare Whelan*, 797 F.2d at 1236 (arguing that everything not necessary to the purpose or function of a work is expression), *with Borland*, 49 F.3d at 815 (stating that methods of operation are means by which a user operates something and any words used to effectuate that operation are unprotected expression).

³⁴ *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1525 (9th Cir.1992) ("In our view, in light of the essentially utilitarian nature of computer programs, the Second Circuit's approach is an appropriate one."); *Mitel*, 124 F.3d at 1372.

³⁵ *Borland*, 49 F.3d at 815, n.8 (recognizing that *Altai* never states that every work contains a copyrightable "nugget" of protectable expression but expressing concern that any literal copying would by definition be copying expression).

³⁶ *Altai*, 982 F.2d at 702 (citing *Whelan*, 797 F.2d, at 1233); *Stern Electronics, Inc. v. Kaufman*, 669 F.2d 852, 855 n.3 (2d Cir. 1982); *Franklin*, 714 F.2d at 1246-47 (discussing source and object code).

³⁷ *Lexmark*, 387 F.3d at 533. ("[S]ource code...[is] the spelled-out program commands that humans can read.")

³⁸ *Altai*, 982 F.2d at 698 ("Object code is the binary language comprised of zeros and ones through which the computer directly receives its instructions.")

³⁹ *Phoenix Control*, 886 F.2d at 1175.

⁴⁰ *Id.*

⁴¹ *Borland*, 49 F.3d at 815.

⁴² *See Sony Computer Ent'mt, Inc. v. Connectix Corp.*, 203 F.3d 596, 599-600 (9th Cir. 2000).

ibility are not eligible for copyright.⁴³

III. THE FEDERAL CIRCUIT'S DECISION

The Federal Circuit, reviewing *de novo* and applying the law of the Ninth Circuit⁴⁴, found that Oracle's API packages were copyrightable.

A. Process, System, or Method of Operation

The Federal Circuit held that copyrightability was not barred because it did not find the API code to fall within the 102(b) preclusions for a process, system, or method of operation. Patents protect the process or method performed by a computer program but copyrights protect *the expression* of that process or method.⁴⁵

Under Oracle's theory, the uncopyrightable method of operation or system or process is the underlying computer function triggered by the declaring code while the code itself is eligible for copyright protection.⁴⁶ Google argues that the copying is necessary to ensure compatibility and interoperability, furthering innovation.⁴⁷ Based on the assumption that interface specifications were uncopyrightable, the free and open use of APIs has been both routine and essential in the computer industry since its beginning.⁴⁸ The assumption of uncopyrightability was based on *Lotus*. APIs appear similar to the menu hierarchy addressed in *Lotus*.

In dismissing the Ninth Circuit's holding in *Sega* that functional requirements for achieving capability are not copyrightable⁴⁹, the Federal Circuit essentially declared that *Sega* was only a fair use case.⁵⁰ Yet, the court in *Sega* held specifically that copying exact lines of code was non-infringing when that code was essential to achieving interoperability.⁵¹ If it is essential to achieving interoperability, then the program must be a method of operation or system or

⁴³ *Sega*, 977 F.2d at 1522.

⁴⁴ *Google II*, 750 F.3d at 1353, n.3 (observing that the U.S. Supreme Court has not stated the precise standard of review for copyrightability appeals) (citing *Ets-Hokin v. Sky Spirits, Inc.*, 225 F.3d 1068, 1073 (9th Cir.2000)).

⁴⁵ *Atari*, 975 F.2d at 839 (emphasis added).

⁴⁶ Brief of the United States as Amicus Curiae at 14, *Google, Inc. v. Oracle America, Inc.*, No. 14-410 (U.S. June 29, 2015).

⁴⁷ *Id.*

⁴⁸ See Brief of Computer Scientists as Amicus Curiae Supporting Petitioner at 5, *Google, Inc. v. Oracle America, Inc.*, No. 14-410 (U.S. June 29, 2015).

⁴⁹ *Sega*, 977 F.2d at 1526.

⁵⁰ *Google II*, 750 F.3d at 1369.

⁵¹ *Sega*, 977 F.2d at 1522-24.

process. Expression could not be essential to achieving interoperability. Rather, it is the system, process of commands, and quite literally the method of operation within an API that necessitates copying for purposes of compatibility.

APIs are designed to be inherently functional and achieve only technical goals of efficiency.⁵² Oracle is seeking through its copyright claim to obtain an exclusive right to a functional system, process, or method of operation that should instead be protected, if at all, by a patent.⁵³ The Federal Circuit too easily disregarded the practical considerations of fostering compatibility; in doing so, it missed an opportunity to find that the 102(b) preclusion of copyrightability for a process, method, or system of operation applies. Ending the application of 102(b) to APIs would have chilling consequences on innovation within startup firms and on open source developers creating programs for existing platforms and interfaces.⁵⁴

B. Structure, System, and Organization

The Federal Circuit found Google infringed upon the “structure, sequence, and organization” of the Java APIs because Google could have organized API packages in several different ways and because Java developers could have organized API packages in any number of ways. Essentially, the court constructively imparts expression into a computer program absent a finding that the idea and expression have merged and a monopoly has been formed. In *Altai*, the Second Circuit found, in the interest of interoperability, that the accused programs were “dictated by the functional demands” of the programs at issue in order to provide the same services and to be compatible with the original program.⁵⁵ The court’s abstraction-filtration-comparison test filters out the structural elements of programs such as efficient design elements, elements constrained by external factors, and standard programming techniques.⁵⁶ Seemingly, all three of those would apply to Java APIs, leaving nothing copyrightable.⁵⁷ Therefore, the abstraction-filtration-comparison test is more consistent with Ninth Circuit precedent and is a more precise test than the SSO test in conceptualizing the idea/expression dichotomy.

⁵² *Id.* at 1524

⁵³ *Google II*, 750 F.3d at 1353.

⁵⁴ Brief of Intellectual Property Professors as Amicus Curiae Supporting Grant of Petition at 24, *Google Inc., v. Oracle America, Inc.*, No. 14-410 (U.S. June 29, 2015).

⁵⁵ *Altai*, 982 F.2d at 706

⁵⁶ *Id.* at 707-10.

⁵⁷ *Id.* at 704.

C. Other Barriers to Copyrightability

The Federal Circuit held that copyrightability was not barred by the merger doctrine. If the expressed idea can only be expressed in one way, then the idea and expression have “merged”, precluding copyright.⁵⁸ The court adopted the computer program-specific Ninth Circuit position in *Altai* that a “unique arrangement of computer program expression...does not merge with the process so long as alternate expressions are available.”⁵⁹ To the Federal Circuit, Java creators were not selecting from pre-ordained names and phrases in writing the declaring code. Google could have chosen different language to do the same thing in its declaration code so the merger doctrine did not apply.⁶⁰ The District Court erred in applying the merger doctrine because the merger doctrine is not meant to consider whether Google had choices in whether or not to use the same names to achieve the same functionality.⁶¹ Google had to use the same words to reap the benefits of Java.⁶² However, the merger doctrine applies to the person writing the original code, not the copier. The Java programmers had “unlimited options as to the selection and arrangement of the 7000 lines Google copied.”⁶³ The court did recognize that in the case of the three core packages, the Java authors may have had a limited number of ways to express the methods within the Java language. If that were the case, merger would apply to any copying of those packages.

The merger doctrine precludes the copyright of the only way of saying something.⁶⁴ The Federal Circuit’s requirement that the merger of the idea and expression occur at the original creation is problematic. The idea of program compatibility is that the basics are always the same so that way software can be used in more than one place. Java created and offers the API packages in the public domain precisely to create a universal language of programming that fosters compatibility.⁶⁵ At the time the thirty-seven packages were written, it was Java’s idea that they would be used by programmers in the development of software. There could have been any number of ways to write the language

⁵⁸ *Satava v. Lowry*, 323 F.3d 805, 812 n. 5 (9th Cir.2003).

⁵⁹ *Atari*, 975 F.2d at 840 (citing *Apple Comput. v. Formula Int’l* 725 F.2d 521, 525 (9th Cir. 1984)).

⁶⁰ *Google II*, 750 F.3d at 1361-62.

⁶¹ *Id.*

⁶² *Id.*

⁶³ Brief for Appellant, *supra* note 1, at 50.

⁶⁴ *Google II*, 750 F.3d at 1359.

⁶⁵ *What is Java Technology and Why Do I Need It?*, JAVA (October 27, 2015), https://java.com/en/download/faq/whatis_java.xml (“Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day.”).

itself, but the copying of the language is necessary for it to serve its purpose of compatibility.⁶⁶ The Federal Circuit's logic that any words could have been used in the source code could have been is too limiting. An analysis of the purpose of Java is necessary. That analysis reveals, for the sake of compatibility, there really was no other choice by Google, if it was to make itself available to programmers using Java, then copy the packages verbatim. Therefore, Oracle is attempting to copyright the only way something could have been done and the idea and the expression have merged. That being said, the Federal Circuit appears to correctly point to the appropriate Ninth Circuit precedent indicating that merger is a defense to infringement rather than a barrier to copyrightability.⁶⁷

The Federal Circuit held that copyrightability was not barred by the short phrases doctrine.⁶⁸ The short phrases doctrine prevents copyrights for "words and short phrases such as names, titles, and slogans; mere variations of typographic ornamentation...mere listing of ingredients or contents."⁶⁹ The district court erred by not considering 1) that short words in an original combination can be copyrightable and 2) the creativity of the words and phrases. The short phrases used in Java language must be looked at as a whole as 7,000 lines. Java writers exhibited creativity in the compilation, selection, coordination, and arrangement of the words.⁷⁰

The short phrases doctrine precludes the copyright of the only way of saying something. Computer programming requires very precise language. When source code is entered to create implementing code, there are certain words that must mean certain things.⁷¹ In the case of Java's API packages, certain commands necessarily contain certain words.⁷² However, the short phrases used in the API source code are commands.⁷³ Those commands were chosen to

⁶⁶ Eric N. Shapiro, *Java, Not Swift, Will Be The Universal Language for Cross-Platform Mobile App Development*, ARCTOUCH (Nov. 22, 2015), <http://arctouch.com/2015/10/java-cross-platform-mobile-app-development> (stating the abundance of Java developers and Java's maturity making it preferable for cross-platform application development).

⁶⁷ *Ets-Hokin*, 225 F.3d at 1082 (9th Cir. 2000) (citing *Kregos v. Associated Press*, 937 F.2d 700, 705 (2d Cir.1991)); see also *Hart v. Dan Chase Taxidermy Supply Co., Inc.*, 86 F.3d 320, 322 (2d Cir. 1996) (holding that there is a "strong preference that the question [of merger] be decided only after all the evidence of substantial similarity is before the court.")

⁶⁸ *Google II*, 750 F.3d at 1363.

⁶⁹ 37 C.F.R. § 202.1.

⁷⁰ *Atari*, 975 F.2d at 840; see also 17 U.S.C. §§ 101, 103 (recognizing copyright protection for "compilations" which are defined as work that is "selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship").

⁷¹ *Google I*, 872 F.Supp.2d at 977.

⁷² *Id.* at 977-78.

⁷³ *Id.*

be arranged in a specific way to achieve a desired result.⁷⁴ Therefore, the Federal Circuit's decision to view the 7,000 lines of text as a whole is appropriate.

The Federal Circuit held copyrightability was not barred by the scenes a faire doctrine.⁷⁵ The scenes a faire doctrine bars expression that is "standard, stock, or common to a topic."⁷⁶ Its rationale deriving from both the merger doctrine and the idea/expression dichotomy, expressions are basically ideas when they are indispensable and naturally associated with the treatment of a given idea.⁷⁷ The Federal Circuit upheld the previously established precept that the scenes a faire doctrine is a defense to infringement rather than a barrier to copyrightability in the Ninth Circuit.⁷⁸

The Java code would contain commands that are common to programming but would not seemingly be indispensable and naturally associated. The court cited one of the API packages titled "java.lang."⁷⁹ The declaring code for the max method within the math class is "public static int max(int x, int y)". That looks like declaring code, but that language is not indispensable to programming. That language is not necessary to all forms of code. Therefore, the language in the APIs would not be precluded as scene a faire.⁸⁰

IV. CONCLUSION

The Federal Circuit, finding that there was insufficient evidence to render a judgment on the issue of the application of the Fair Use doctrine, remanded that issue back to the district court. Like in *Sega*, the court will have to analyze whether compatibility or interoperability represents, as a matter of law, fair use of copyrightable code. With the Supreme Court having refused to resolve the circuit split on the issue of copyrightability, the concerns regarding participation, competition, reimplementing, and the future of open source software linger and would seemingly now only be resolved by a granting of the fair use exception. Barring that, licensing fees and concern over litigation will undoubtedly slow innovation because programs will be kept from building on each other because each will need to survive a test of originality. If none of the

⁷⁴ *Id.*

⁷⁵ *Google II*, 750 F.3d at 1364.

⁷⁶ *Mitel*, 124 F.3d at 1374; *see generally* 4-13 Nimmer on Copyright § 13.03[B][4].

⁷⁷ *Swirsky v. Carey*, 376 F.3d 841, 850 (9th Cir. 2004).

⁷⁸ *Google II*, 750 F.3d at 1364 (citing 4-13 Nimmer on Copyright § 13.03[B][3])13; *see also* 4-13 Nimmer on Copyright § 13.03[B][4] ("[T]his doctrine does not limit the subject matter of copyright; instead, it defines the contours of infringing conduct.").

⁷⁹ *Google II*, 750 F.3d at 1349, n.2.

⁸⁰ *Id.* at 1363-64 (explaining that though programmers may be accustomed to certain API packages, scenes a faire doctrine is not a defense to copyrightability in regard to coding in general, so it would follow that scenes a faire would not apply to API packages).

102(b) exceptions apply, the finding of originality and literal copying of a computer program is essentially dispositive of copyrightability. The consequences of the Federal Circuit's decision will likely be felt by developers, by consumers, and by the technology industry as a whole for decades to come.