No. 14-410

In the Supreme Court of the United States

GOOGLE, INC., PETITIONER

v.

ORACLE AMERICA, INC.

ON PETITION FOR A WRIT OF CERTIORARI
TO THE UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

BRIEF FOR THE UNITED STATES AS AMICUS CURIAE

DONALD B. VERRILLI, JR.
Solicitor General
Counsel of Record
BENJAMIN C. MIZER
Principal Deputy Assistant
Attorney General
MALCOLM L. STEWART
Deputy Solicitor General
JOHN F. BASH
Assistant to the Solicitor
General
DOUGLAS N. LETTER
MARK R. FREEMAN
SONIA K. MCNEIL
Attorneys

Department of Justice
Washington, D.C. 20530-0001
SupremeCtBriefs@usdoj.gov
(202) 514-2217

QUESTION PRESENTED

The Copyright Act of 1976, 17 U.S.C. 101 *et seq.*, protects "original works of authorship," 17 U.S.C. 102(a), including "computer program[s]," 17 U.S.C. 101. The Act specifies, however, that copyright protection does not "extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. 102(b). The question presented is as follows:

Whether Section 102(b) precludes copyright protection for original software code that defines and organizes a set of functions that are useful in writing computer programs.

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Cases:

Baker v. *Selden*, 101 U.S. 99 (1880)....................12, 14, 16, 20

Campbell v. *Acuff-Rose Music, Inc.*, 510 U.S. 569 (1994) 3

Feist Publ'ns, Inc. v. *Rural Tel. Serv. Co.*, 499 U.S. 340 (1991) 2, 9, 11

Golan v. *Holder*, 132 S. Ct. 873 (2012).............................2, 12

Harper & Row Publishers, Inc. v. *Nation Enters.*, 471 U.S. 539 (1985) 17

Lexmark Int'l, Inc. v. *Static Control Components, Inc.*, 387 F.3d 522 (6th Cir. 2004) 17, 19, 21

Lotus Dev. Corp. v. *Borland Int'l, Inc.*, 49 F.3d 807 (1st Cir. 1995), aff'd by an equally divided court, 516 U.S. 233 (1996) 19, 20, 21

Sega Enters. Ltd. v. *Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992)...18

Situation Mgmt. Sys. v. *ASP Consulting LLC*, 560 F.3d 53 (1st Cir. 2009)...21

Sony Computer Entm't, Inc. v. *Connectix Corp.*, 203 F.3d 596 (9th Cir.), cert. denied, 531 U.S. 871 (2000)......17

Twin Peaks Prods., Inc. v. *Publications Int'l, Inc.*, 996 F.2d 1366 (2d Cir. 1993)..2

Cases—Continued: Page

Statutes:

Miscellaneous:

In the Supreme Court of the United States

No. 14-410

GOOGLE, INC., PETITIONER

v.

ORACLE AMERICA, INC.

ON PETITION FOR A WRIT OF CERTIORARI
TO THE UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

BRIEF FOR THE UNITED STATES AS AMICUS CURIAE

This brief is submitted in response to the Court's order inviting the Solicitor General to express the views of the United States. In the view of the United States, the petition for a writ of certiorari should be denied.

STATEMENT

1. a. The Copyright Act of 1976, 17 U.S.C. 101 *et seq.*, provides that "[c]opyright protection subsists * * * in original works of authorship." 17 U.S.C. 102(a). Works of authorship include, as relevant here, "literary works," which are "works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia." 17 U.S.C. 101, 102(a)(1). To be "original" in the relevant sense, a work must have been "independently created by the author (as opposed to copied from other works)" and must "possess[] at least some minimal degree of crea-

tivity." *Feist Publ'ns, Inc.* v. *Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991). A copyright extends not only to the literal aspects (*i.e.*, the actual text) of an original literary work, but also to its non-literal aspects, such as the plot of a novel, to the extent they are original. See, *e.g.*, *Twin Peaks Prods., Inc.* v. *Publications Int'l, Inc.*, 996 F.2d 1366, 1372-1373 (2d Cir. 1993).

The scope of the rights that a copyright confers is subject to limitations set out in the Copyright Act. As particularly relevant here, the protection afforded by a copyright in a work does not "extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. 102(b). For example, a copyright for a book that explains how to perform a new surgical method would bar others from copying the book, but not from practicing the method that the book describes. Section 102(b) codifies the longstanding common-law principle known as the "idea/ expression dichotomy." *Golan* v. *Holder*, 132 S. Ct. 873, 890 (2012); see H.R. Rep. No. 1476, 94th Cong., 2d Sess. 57 (1976).

Other common-law doctrines impose limitations on the copyrightability of expressive works. For example, under the "merger" doctrine, if an idea "can only be expressed in a limited number of ways," those means of expression "cannot be protected, lest one author own the idea itself." *Zalewski* v. *Cicero Builder Dev., Inc.*, 754 F.3d 95, 102-103 (2d Cir. 2014). In that circumstance, the idea and the expression are said to "merge." Similarly, "the doctrine of 'scènes-à-faire' teaches that elements of a work that are indispensable, or at least standard, in the treatment of a given topic—like cowboys, bank robbers, and shootouts in stories of the American West—get no protection." *Id.* at 102 (internal quotation marks omitted).

b. A copyright subsists in a work for a term prescribed by the Copyright Act. 17 U.S.C. 302. A valid copyright gives the owner certain "exclusive rights," including the right "to reproduce the copyrighted work" and "to prepare derivative works based upon the copyrighted work." 17 U.S.C. 106(1) and (2). The Copyright Act also places certain limitations on the enforcement of those exclusive rights.

One such limitation is the "fair use" doctrine, see 17 U.S.C. 107, a "judge-made doctrine" that Congress codified in 1976. *Campbell* v. *Acuff-Rose Music, Inc.*, 510 U.S. 569, 576 (1994). The fair-use doctrine permits certain uses of a copyrighted work when enforcing a copyright would "stifle the very creativity which [copyright] law is designed to foster." *Id.* at 577 (citation and internal quotation marks omitted). The doctrine helps to resolve "the inherent tension in the need simultaneously to protect copyrighted material and to allow others to build upon it." *Id.* at 575. Section 107 identifies a non-exclusive list of factors that are relevant to whether a particular use of a copyrighted work constitutes "fair use." Those factors include "the purpose and character of the use," "the nature of the copyrighted work," and "the amount and substantiality of the portion used in relation to the copyrighted work as a whole." 17 U.S.C. 107(1)-(3).

c. This case concerns the copyrightability of computer code. To induce a computer to perform a function, a person must give the computer written instructions. Typically, those instructions are written in "source code," which consists of words, numbers, and symbols in a particular "programming language," which has its own unique syntax and semantics. The source code is then converted into binary "object code"—ones and zeros—that is readable by the computer.

It is both well-settled and undisputed in this case that computer code can be copyrightable as a "literary work." 1 Melville B. Nimmer and David Nimmer, *Nimmer on Copyright* § 2.04(C)(2) (2013). Section 101 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." A number of Copyright Act provisions recognize that a person may own a copyright in a "computer program." See, *e.g.*, 17 U.S.C. 109(b)(1)(A), 117, 506(a)(3)(A).

2. a. Respondent's predecessor-in-interest created the Java programming language. Pet. App. 4 & n.1, 104-106. It then developed a variety of tools and software—known collectively as the "Java platform"—to assist programmers in writing and distributing computer programs in Java. *Id.* at 101.

Like many programming languages, Java allows programmers to use short, modular subprograms to create longer, more complex computer programs. For example, in creating a video game, a programmer might create subprograms to perform various discrete tasks, such as displaying text on the screen or playing a sound. In Java, these subprograms are called "methods." As part of the Java platform, respondent provides to computer programmers a "Standard Library" of thousands of prewritten methods, which have been organized into "classes" and "packages." See Pet. App. 5-6. The Java version at issue in this case includes 166 packages, comprising more than 600 classes and more than 6000 methods. See *id.* at 6, 106. Although a Java programmer is free to write new code from scratch without relying on prewritten methods, the Java Standard Library provides convenient building blocks for writing computer programs.

Although respondent does not claim a copyright interest in the Java language itself, respondent owns a copyright in the Java Standard Library. Pet. App. 7. Respondent makes the Java Standard Library available to computer programmers under any of three copyright licenses, including a conditional royalty-free license. See *id.* at 7-8.

b. In general, to create a new Java method, a programmer must write software code that tells the computer both (i) what the method is, including its name, the circumstances in which it should be available to programmers, what types of input data it should accept, what types of output data it should produce, and what types of errors it can generate; and (ii) how to perform the method, including steps for using the specified input data to produce the specified type of output data. The parties have referred to the first type of code as "declaring code" and to the second as "implementing code." See Pet. App. 6-7, 106-109, 111, 113-115.

An example drawn from the district court's opinion illustrates the distinction. See Pet. App. 113-114. The following Java code defines a method named "max" that returns the larger of two integers, *x* and *y*:

```
Line 1:  public static int max (int x, int y) {
Line 2:  if (x > y) return x;
Line 3:  else return y;
Line 4:  }
```

In this example, Line 1 is the "declaring code," which tells the computer the name of the method (*max*); the circumstances in which the method is available to programmers (*public* and *static*); the type of output data it produces (*int*); and the type and order of the input data it accepts (integer *x* and integer *y*). (Additional declaring code not reproduced above places the method into a class, and the class into a package.) Lines 2-4 are the "implementing code," which instructs the computer how to use the input data to produce the output data.

Once a method has been written, a programmer may invoke the method at any time by typing a command consisting of the name of the method (and possibly its package and class) and the appropriate input data. Although that command is determined by the method's declaring code, it is not identical to the declaring code. As a result of this structure, to use the prewritten methods in respondent's Java Standard Library, a programmer need not see or understand the implementing code; the programmer need only know (or look up) the name of the relevant method and the parameters established by its declaring code.

c. Petitioner developed the Android operating system for mobile devices. Petitioner also created its own platform—*i.e.*, a set of programming tools—to assist others in developing applications for Android. Pet. App. 8-9. The Android platform uses the Java programming language, but petitioner purposely designed Android not to be compatible with the Java platform or interoperable with Java programs. *Id.* at 9, 56.

Like the Java platform, petitioner's Android platform contains a collection of prewritten methods organized into classes and packages. Pet. App. 9. Petitioner created much of the Android library from scratch. For 37 of the 168 packages included in the Android library, however, petitioner copied the Java declaring code verbatim, while writing its own implementing code. *Ibid.* The copied packages contained the Java methods and classes that petitioner concluded would be most useful for developing smartphone applications. See Pet. 31. Petitioner asserts that it copied the declaring code so that programmers familiar with the Java platform would be able to switch over to the Android platform without having to learn entirely new commands for invoking commonly used methods.

In total, petitioner copied approximately 7000 lines of declaring code. Pet. App. 14-15. In so doing, petitioner also copied the complex architecture of the 37 packages at issue, including the names and specifications of the methods and classes in those packages and their hierarchical and interdependent relationship to each other. *Ibid.*

3. In August 2010, respondent filed suit against petitioner in the United States District Court for the Northern District of California. Respondent alleged, *inter alia*, that petitioner had infringed its copyright in the Java Standard Library. Am. Compl. ¶¶ 37-40, Ex. H. The parties and the courts below treated the complaint as alleging two distinct theories of copyright infringement: (a) literal, verbatim copying of respondent's declaring code; and (b) non-literal copying of the "structure, sequence, and organization" (SSO) of the Java Standard Library, which the declaring code establishes and reflects.

As relevant here, petitioner argued that the SSO constitutes a "system" or "method of operation" that is ineligible for copyright protection under Section 102(b) of the Copyright Act. Petitioner also asserted a fair-use defense under Section 107. The parties agreed that a jury would decide infringement and fair use, and that the judge would decide whether the code was copyrightable. Pet. App. 2. The jury found infringement, but hung on fair use. *Id.* at 12.

The district court then set aside the infringement verdict on the ground that respondent did not possess a valid copyright interest in the material that petitioner had copied. Pet. App. 3. The court agreed with respondent that the declaring code and the SSO of the Java Standard Library reflect sufficient creativity and originality to support copyright protection under Section 102(a). *Id.* at 22. The court held, however, that under Section 102(b), the SSO is ineligible for copyright protection because it constitutes a "method of operation" or "system" for using the prewritten subroutines included in the Java platform. *Id.* at 158-159. The court also held that the declaring code was not copyrightable under the merger doctrine, but rejected petitioner's invocation of the *scènes à faire* doctrine as applied to the SSO. *Id.* at 154-157 & n.9.

4. The United States Court of Appeals for the Federal Circuit reversed. Pet. App. 1-78.[1] The court first explained that, "[a]t this stage, it is undisputed that the declaring code and the [SSO] of the Java * * * packages are original" and thus meet Section 102(a)'s basic requirement for copyrightability. *Id.* at 22; see Pet. C.A. Br. 29-30. As the court understood

[1] The Federal Circuit had appellate jurisdiction because of patent claims not relevant here. See 28 U.S.C. 1295(a)(1).

the dispute, the parties principally disagreed over "the proper interpretation and application of Section 102(b)," with petitioner arguing that, even if a work is original, "Section 102(b) takes [copyright protection] away if the work has a functional component." Pet. App. 23.

The court of appeals rejected that argument, explaining that Section 102(b) "restate[s] * * * the basic dichotomy between expression and idea." Pet. App. 23 (quoting *Feist*, 499 U.S. at 356); see *id.* at 41-49. The court concluded that, for these purposes, computer code is "expression" despite its functional character. The court explained that, because "computer programs are by definition functional" in that "they are all designed to accomplish some task," the district court's analysis implied that "no computer program is protectable," a result that would "contradict[] Congress's express intent to provide copyright protection to computer programs." *Id.* at 46.

The court of appeals rejected petitioner's invocation of the merger doctrine. The court explained that the evidence presented at trial "showed that [respondent] had 'unlimited options as to the selection and arrangement of the 7000 lines [petitioner] copied.'" Pet. App. 33 (quoting Resp. C.A. Br. 50). The court noted that petitioner did not contend that any particular package could have been written in only a limited number of ways. *Id.* at 35 & n.8. The court of appeals also found the *scènes à faire* doctrine to be inapplicable, noting that petitioner had "not objected to the trial court's conclusion that [petitioner] failed to make a sufficient factual record" to support that argument. *Id.* at 38-40 (citation omitted).

Finally, the court of appeals addressed petitioner's argument that, because "software developers were already trained and experienced in using the Java * * * packages at issue[,]" copying was necessary to make its technology interoperable with respondent's technology. Pet. App. 57. The court explained that, although "this competitive objective might be relevant to the fair use inquiry," it did not bear on the threshold question of copyrightability. *Ibid.* The court remanded for a new trial on fair use after finding that the record lacked "sufficient factual findings" for that issue to be resolved as a matter of law. *Id.* at 69.

DISCUSSION

Petitioner does not dispute that the declaring code that it copied verbatim from respondent's Java Standard Library reflected significant creative choices and is part of an "original work[] of authorship" under Section 102(a) of the Copyright Act. See Pet. App. 23. Petitioner contends, however, that the declaring code is not entitled to copyright protection because it constitutes a "method of operation" or "system" under Section 102(b). That argument lacks merit.

Despite the inherently functional character of all computer code, the Copyright Act makes clear that such code can be copyrightable. Nothing about the declaring code at issue here materially distinguishes it from other computer code, and petitioner has identified no genuine conflict of authority concerning Section 102(b)'s applicability to circumstances like these. Although petitioner has raised important concerns about the effects that enforcing respondent's copyright could have on software development, those concerns are better addressed through petitioner's fair-use defense, which will be considered on remand.

Accordingly, this Court's interlocutory review is not warranted.

A. The Court Of Appeals Correctly Held That Section 102(b) Does Not Foreclose Copyright Protection For Respondent's Declaring Code

Various Copyright Act provisions make clear that a person may own a copyright in a "computer program," which is defined 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; see p. 4, *supra*. For example, Section 117, entitled "Limitations on exclusive rights: Computer programs," sets out the circumstances in which "the owner of a copy of a computer program" may "make or authorize the making of another copy" without infringing the copyright in the program. 17 U.S.C. 117(a). Computer code also fits comfortably within the statutory definition of a "literary work" because it is a work "expressed in words, numbers, or other verbal or numerical symbols or indicia." 17 U.S.C. 101; see H.R. Rep. No. 94-1476, 94th Cong., 2d Sess. 51 (1976).

To be copyrightable, any particular computer code must meet the basic requirements of copyright law. The foremost requirement is that the work be "original," 17 U.S.C. 102(a), *i.e.*, that it "possess[] at least some minimal degree of creativity." *Feist Publ'ns, Inc.* v. *Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991). Computer code is also subject to various traditional limitations, such as the merger and *scènes à faire* doctrines. See pp. 2-3, *supra*.

Those requirements, however, are not currently at issue in this case. Petitioner conceded below that respondent had exercised creative choices in naming and arranging the methods in the Java Standard Library. See Pet. App. 22. And petitioner has not sought review of the court of appeals' holdings with respect to various traditional copyright limitations. See *id.* at 30-41.

Petitioner contends, however, that even if the declaring code is an "original work[] of authorship" under Section 102(a), it is not entitled to copyright protection because it constitutes a "method of operation" or "system" within the meaning of Section 102(b). That argument is incorrect.

1. a. Section 102(b) codifies the "idea/expression dichotomy," a longstanding tenet of copyright law. *Golan* v. *Holder*, 132 S. Ct. 873, 890 (2012); see p. 2, *supra*. That principle means that a copyright in an "original work of authorship" under Section 102(a) covers only the expressive work itself. "In no case does copyright protection * * * extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery" that is "described, explained, illustrated, or embodied" in the work. 17 U.S.C. 102(b). Section 102(b) is not a limitation on what kinds of expressive works may be protected by a copyright. Rather, it is a limitation on how broadly the copyright extends. Although a book on how to build a bicycle may be eligible for copyright protection, that copyright does not include any exclusive right to practice the bicycle-building method that the book explains; nor can the author prevent another person from writing a better book with a clearer explanation of the same process. See *Baker* v. *Selden*, 101 U.S. 99, 105 (1880). Copyright law protects the means of expressing ideas or concepts, but it does not give the copyright holder the right to exclude others from making use of the ideas or concepts themselves.

Petitioner therefore is incorrect in suggesting (see Pet. 26; Pet. App. 23) that a work could be both an "original work[] of authorship" protectable under Section 102(a) *and* a "method of operation" or "system" under Section 102(b). If a work constitutes expression (and if it is original), it is copyrightable under Section 102(a). Section 102(b) merely excludes from copyright protection the subject matter explained or described in the expressive work.

b. If the Copyright Act contained no explicit references to computer code, one might reasonably conclude that such code is not protectable "expression" at all. Computer code differs in a fundamental way from many traditional means of literary expression. A book or newspaper article is meant to be read and comprehended by a human being as a description of an idea or story. Although many copyrightable written documents explain how practical tasks should be performed, there is typically a clear distinction between the written explanation and the actual performance of the task. Computer code, by contrast, is *both* expression *and* the actual means by which a computer is induced to perform the desired function. It therefore would not be unnatural to describe computer code as a "method of operation" or "system." Nor would it be unreasonable to conclude that, as between a book on bicycle-building and the actual construction of a bicycle, computer code is more analogous to the latter.

The Copyright Act as a whole makes clear, however, that the functional character of computer code cannot be sufficient to bring it within Section 102(b). If that were so, no computer code would qualify for copyright protection; yet the Copyright Act unequivocally recognizes that a person can own a copyright in computer code. See p. 11, *supra*. Rather, the uncopyrightable "method of operation" or "system" or "process" is the underlying computer function triggered by the written code—for example, an algorithm that the computer executes to sort a data set. The code itself, however, is eligible for copyright protection.

If within a given technological environment, code must be drafted in a specific way in order to induce the computer to carry out a particular function, then the expression would "merge" with the function, and the code would be uncopyrightable. Cf. *Baker* v. *Selden*, *supra*. And some computer routines may be so standard in the programming industry that the *scènes à faire* doctrine deprives them of copyright protection. But computer code is not an uncopyrightable "method of operation" or "system" under Section 102(b) simply because it causes a computer to function.

2. Petitioner does not contend that all computer code constitutes a "method of operation" or "system." See Pet. 27. Rather, petitioner argues that the declaring code at issue here has a special quality that distinguishes it from computer code in general. According to petitioner (Pet. 29-32), because the declaring code dictates the commands that Java programmers must use to invoke prewritten methods, the declaring code is the "method of operat[ing]" those methods, whereas the implementing code is not.

That distinction does not withstand scrutiny. Both declaring code and implementing code ultimately perform the same practical function: They instruct a computer to work. The declaring code tells the computer to call up the implementing code, and the implementing code tells the computer to perform an operation, such as executing a sorting algorithm.

Both are necessary components of a Java or Android method. And neither the declaring code nor the implementing code is what a programmer physically types when invoking a method. See p. 6, *supra*.

Declaring code may be one step further removed than implementing code from the ultimate operation that a computer performs. Petitioner offers no conceptual or linguistic basis, however, for concluding that the status of particular code as a Section 102(b) "method of operation" or "system" depends on the directness of the link between that code and the ultimate computer function. And even if the terms "method of operation" and "system" were otherwise susceptible of such a reading, the statutory definition of "computer program"—"a set of statements or instructions to be used *directly or indirectly* in a computer in order to bring about a certain result"—indicates that copyrightability should not turn on that distinction. 17 U.S.C. 101 (emphasis added).

Petitioner's Section 102(b) argument also suffers from a broader flaw. Respondent owns a copyright in the Java Standard Library, which includes both declaring code and implementing code. The declaring code is thus a segment of a much larger "original work of authorship." 17 U.S.C. 102(a). The basic purpose of Section 102(b), however, is not to distinguish between copyrightable and uncopyrightable portions of a larger work of authorship, but rather to distinguish between the work of authorship itself and something else—be it an idea, a process, or a method of operation—that the work of authorship describes or explains. Thus, while it may be sensible to distinguish between declaring code and implementing code for other copyright-law purposes (such as fair use, see pp.

17-19, *infra*), it would be anomalous to draw that distinction in applying Section 102(b).

Petitioner's reliance (Pet. 22-27) on *Baker* v. *Selden*, *supra*, is misplaced. *Baker* involved a copyrighted book that explained a system of accounting and included forms that could be used to implement the system. The Court held that the copyright in the book itself did not bar others from using substantially similar forms to practice the accounting method that the book described. 101 U.S. at 101, 107. Critical to the Court's analysis, however, was that the accounting method could not be practiced other than through forms like the ones reprinted in the book. See *id.* at 104-105. The Court thus decided the case based on what was effectively a merger analysis. Here, by contrast, petitioner does not dispute the court of appeals' statement that there were "unlimited" ways that respondent could have named and structured its methods, Pet. App. 33, and nothing logically required petitioner to copy respondent's declaring code when it created the Android platform.

3. Petitioner contends (Pet. 33-37) that declaring code should be classified as an uncopyrightable "method of operation" or "system" under Section 102(b) in order to ensure that copyright law does not impede the "compatibility" or "interoperability" of different computer systems. Petitioner further contends that, if copyright holders like respondent can prevent the copying of declaring code in these circumstances, they can exercise long-term "monopolies on the basic building blocks of innovation" in the software industry. Pet. 24. Petitioner does not and could not plausibly assert, however, that interoperability with the Java platform is a prerequisite to Android's

proper functioning. To the contrary, petitioner "designed Android so that it would *not* be compatible with the Java platform." Pet. App. 56 (emphasis added). Rather, petitioner argues that programmers have become fluent in respondent's Java platform; that they will be deterred from writing programs for Android if they are required to learn all new commands; and that verbatim copying of respondent's declaring code was necessary for the familiar commands to work on the Android platform. See *id.* at 57.

The general concerns that petitioner raises are substantial and important, but Section 102(b) is not the appropriate statutory provision to address them. Rather, legitimate concerns with interoperability and lock-in effects are far better addressed through the fair-use doctrine codified at Section 107. Petitioner argues that its copying of respondent's code promoted innovation by enabling programmers to switch more easily to another platform. But it is the function of the fair-use doctrine, not of Section 102(b), to identify circumstances in which the unauthorized use of copyrighted material will promote rather than disserve the purposes of the copyright laws. The doctrine is an "equitable rule of reason" that "permits courts to avoid rigid application of the copyright statute when, on occasion, it would stifle the very creativity which that law is designed to foster." *Harper & Row Publishers, Inc.* v. *Nation Enters.*, 471 U.S. 539, 550 n.3 (1985) (citations omitted). Interoperability and lock-in concerns like those raised by petitioner can appropriately be considered as part of fair-use analysis. See *Lexmark Int'l, Inc.* v. *Static Control Components, Inc.*, 387 F.3d 522, 544-545 (6th Cir. 2004); *Sony Computer Entm't, Inc.* v. *Connectix Corp.*, 203 F.3d 596, 602-605 (9th Cir.), cert. denied, 531 U.S. 871 (2000); *Sega Enters. Ltd.* v. *Accolade, Inc.*, 977 F.2d 1510, 1523-1524 (9th Cir. 1992).[2]

Indeed, many of petitioner's specific contentions will be relevant to its fair-use defense on remand. For example, although it would be anomalous to use Section 102(b) to distinguish between different segments of a single work of authorship (see pp. 15-16, *supra*), Section 107(3) instructs courts to consider "the amount and substantiality of the portion [of a copyrighted work] used in relation to the copyrighted work as a whole" in adjudicating a fair-use defense. That petitioner copied only respondent's declaring code while writing its own implementing code should therefore be a relevant factor in the lower courts' fair-use analysis. See Pet. App. 68. The lower courts likewise can take into account petitioner's contention (Pet. 33) that "the ability to build on existing interfaces in creating new products and services is a critical driver of innovation in the computer and software fields." By the same token, the lower courts can take into account the precise nature of the "interface" that petitioner sought to foster, *i.e.*, the fact that petitioner's objec-

[2] The fact that a particular computer program or line of computer code has become well known and popular among programmers cannot change its fundamental character from an "original work of authorship" (or segment thereof) to a "method of operation" or "system." 17 U.S.C. 102. If Java's increased popularity could have that effect, the copyrightability of a particular work would turn on events that substantially postdated the work's creation. That result is at odds with the Copyright Act's basic design, under which copyright protection subsists from the creation of a work through the prescribed statutory term. 17 U.S.C. 302. By contrast, a focus on the circumstances that exist at the time of copying is typical of fair-use analysis.

tive was to make the Android platform more attractive to Java-fluent *programmers*, not to make it compatible with the Java *platform*. See p. 17, *supra*; 17 U.S.C. 107(1) (identifying "the purpose and character of the use" as one factor bearing on the fair-use analysis).

Unlike the flexible fair-use doctrine, which considers a broad range of factors, including the purposes for which particular copying is done, Section 102(b) draws a definitive line between copyrightable expression and uncopyrightable subject matter. Petitioner's argument is in substance that, because the copying of declaring code *will often* further the public interest, such code should be categorically unprotected, regardless of the actual purposes for which particular copies are made or the likely effect of particular copying. That approach would not allow courts to consider the full range of competing equities that are implicated by cases like this one.

B. This Court's Review Is Not Warranted

For the foregoing reasons, the court of appeals correctly held that respondent has an enforceable copyright interest in the software code that petitioner copied. Petitioner has identified no sound basis for further review.

1. a. Petitioner contends (Pet. 13-20) that the Federal Circuit's construction of Section 102(b) conflicts with *Lotus Development Corp.* v. *Borland International, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995), aff'd by an equally divided court, 516 U.S. 233 (1996), and *Lexmark*, *supra*. That argument is incorrect.

i. In *Lotus*, the First Circuit held that an original command structure for a spreadsheet program—that is, a menu hierarchy that a lay user (rather than a computer programmer) used to navigate through the program—constituted a "method of operation" that could not be protected under copyright law. 49 F.3d at 815-818. The court found that the "menu command hierarchy does not merely explain and present [the program's] functional capabilities to the user; it also serves as the method by which the program is operated and controlled." *Id.* at 815. The court distinguished "the underlying computer code" of the spreadsheet program on the ground that, "while code is necessary for the program to work, its precise formulation is not." *Id.* at 816. "In other words," the court explained, "to offer the same capabilities," the defendant "did not have to copy [the plaintiff's] underlying code," but "to allow users to operate its programs in substantially the same way," it "had to copy the [plaintiff's] menu command hierarchy." *Ibid.*

The precise rationale of *Lotus* is not clear. Parts of the opinion purport to rest on the proposition that Section 102(b) can foreclose copyright protection for original expression. See 49 F.3d at 818. But other parts of the opinion seem to apply a principle analogous to the merger doctrine, to the effect that, because there was only one menu hierarchy that would allow users to operate the spreadsheet program in substantially the same way, the menu hierarchy (unlike the underlying code) could not acquire copyright protection. See *id.* at 815-816; cf. *Baker*, 101 U.S. at 104-105. Whatever the rationale of *Lotus*, however, the decision cannot reasonably be read to treat Section 102(b) as applicable to computer code itself, a form of expression that the Copyright Act clearly protects and that the First Circuit took pains to distinguish. Petitioner has not identified a single decision in the two decades since *Lotus*, in the First Circuit or elsewhere, in which Section 102(b) has been invoked to hold that original computer code is not copyrightable.

It is true that the decision below, after distinguishing *Lotus* on three separate grounds, stated that the First Circuit's interpretation of Section 102(b) was "inconsistent with binding [Ninth Circuit] precedent." Pet. App. 43. The court took that view because it understood *Lotus* to establish a "hard and fast rule" that "elements which perform a function can never be copyrightable." *Id.* at 44. That is almost certainly an overreading of the First Circuit's analysis, particularly given the *Lotus* court's recognition that computer code, which is inherently functional, is copyrightable. 49 F.3d at 816 & n.11. In any event, the First Circuit has recently acknowledged that Section 102 simply codifies the idea/expression dichotomy. See *Situation Mgmt. Sys.* v. *ASP Consulting LLC*, 560 F.3d 53, 61 (2009). Any tension between *Situation Management Systems* and *Lotus* can be resolved by the First Circuit. See *Wisniewski* v. *United States*, 353 U.S. 901, 902 (1957) (per curiam).

ii. The Sixth Circuit's decision in *Lexmark* provides even less support for petitioner's view. At issue there was the copyrightability of a short computer program that operated as a technological lock barring reuse of certain printer ink cartridges. 387 F.3d at 529-530. Invoking principles of merger and *scènes à faire*, the Sixth Circuit held that the program at issue was likely not copyrightable because its features were largely dictated by external factors, such as the technical requirements under which the program needed to perform. See *id.* at 533-543.

b. Petitioner briefly contends (Pet. 16, 19-20) that the circuits are divided over whether the merger doctrine bears on a work's copyrightability or instead is a defense to infringement. But the court of appeals held that merger principles have no application in this case, "[r]egardless of when the analysis occurs," because respondent had ample freedom of choice in creating its declaring code. Pet. App. 30; see *id.* at 35 & n.8. Since petitioner does not seek review of that holding, any purported conflict of authority over how to classify the merger doctrine would not support further review.

2. Other factors counsel against further review in this case.

First, the courts below have not yet adjudicated petitioner's fair-use defense. 17 U.S.C. 107; see Pet. App. 69. For the reasons discussed above (see pp. 17-19, *supra*), the important concerns that petitioner raises about interoperability and lock-in effects are better addressed through that doctrine. At a minimum, this Court could better assess and clarify the relevance of those concerns to copyright-law analysis if the Court had before it all potentially relevant statutory arguments.

Second, the decision below has limited precedential value. The Federal Circuit's decision applying Ninth Circuit law would not bind a future Ninth Circuit panel, and it would bind future Federal Circuit panels only in cases arising within the Ninth Circuit.

Third, the character of the copyrighted work at issue makes this case a poor vehicle for the Court to address, for the first time, the application of copyright principles to computer programs. Unlike many of the cases that have been the subject of reported appellate decisions, this case does not involve the copying of code for an ordinary computer program. Rather, petitioner copied portions of a collection of prewritten Java methods distributed as part of a "platform" of programming tools designed to assist programmers in writing other computer programs. As a result, the parties and the courts below have devoted considerable attention to questions—such as the distinction between declaring code and implementing code, the technical significance of various features of the Java Standard Library, and the degree to which Java programmers possess familiarity with respondent's prewritten methods—that may have little significance in more common disputes. The Court's resolution of this case therefore might not cast meaningful light on the proper resolution of more typical copyright-infringement cases involving computer programs.

CONCLUSION

The petition for a writ of certiorari should be denied.

Respectfully submitted.

DONALD B. VERRILLI, JR.
Solicitor General
BENJAMIN C. MIZER
Principal Deputy Assistant Attorney General
MALCOLM L. STEWART
Deputy Solicitor General
JOHN F. BASH
Assistant to the Solicitor General
DOUGLAS N. LETTER
MARK R. FREEMAN
SONIA K. MCNEIL
Attorneys

MAY 2015