No. 18-956

In the Supreme Court of the United States

GOOGLE LLC, 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

NOEL J. FRANCISCO
Solicitor General
Counsel of Record
JOSEPH H. HUNT
Assistant Attorney General
MALCOLM L. STEWART
Deputy Solicitor General
MATTHEW GUARNIERI
Assistant to the Solicitor General
MARK R. FREEMAN
DANIEL TENNY
SONIA M. CARSON
Attorneys

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

QUESTIONS 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). Under the "merger" doctrine, copyright protection also does not apply when an idea can be expressed in only a limited number of ways, such that the expression and idea "merge." Finally, the Copyright Act provides that "the fair use of a copyrighted work * * * is not an infringement of copyright." 17 U.S.C. 107.

The questions presented are as follows:

1. Whether Section 102(b) or the merger doctrine precludes copyright protection for respondent's original computer code, which defines and organizes a set of functions that are useful in writing computer programs.

2. Whether the court of appeals correctly held that no reasonable jury could find that petitioner's verbatim copying of respondent's original computer code into a competing commercial product was fair use.

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Cases:

American Broad. Cos. v. *Aereo, Inc.*, 573 U.S. 431 (2014) 21

Atari Games Corp. v. *Nintendo of Am. Inc.*, 975 F.2d 832 (Fed. Cir. 1992) 19

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

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

Disney Enters., Inc. v. *VidAngel, Inc.*, 869 F.3d 848 (9th Cir. 2017) 19

Feist Publ'ns, Inc. v. *Rural Tel. Serv. Co.*, 499 U.S. 340 (1991) 1, 8, 11

Golan v. *Holder*, 565 U.S. 302 (2012) 2, 12

Harper & Row, Publishers, Inc. v. *Nation Enters.*, 471 U.S. 539 (1985) 18, 22

Cases—Continued: Page

Lexmark Int'l, Inc. v. *Static Control Components, Inc.*, 387 F.3d 522 (6th Cir. 2004) ... 15, 16

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) ... 15, 16

Mitel, Inc. v. *Iqtel, Inc.*, 124 F.3d 1366 (10th Cir. 1997) ... 2

Roth Greeting Cards v. *United Card Co.*, 429 F.2d 1106 (9th Cir. 1970) ... 15

Sega Enters. Ltd. v. *Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1993) ... 19

Situation Mgmt. Sys. v. *ASP Consulting LLC*, 560 F.3d 53 (1st Cir. 2009) ... 16

Sony Computer Entm't, Inc. v. *Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000) ... 19

Sony Corp. of Am. v. *Universal City Studios, Inc.*, 464 U.S. 417 (1984) ... 22

TCA Television Corp. v. *McCollum*, 839 F.3d 168 (2d Cir. 2016), cert. denied, 137 S. Ct. 2175 (2017) ... 18

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

Zalewski v. *Cicero Builder Dev., Inc.*, 754 F.3d 95 (2d Cir. 2014) ... 2, 13

Statutes, regulation, and rule:

Copyright Act of 1976, 17 U.S.C. 101 *et seq.* ... 1

17 U.S.C. 101 ... 1, 4, 11

17 U.S.C. 102(a) ... 1, 8, 11, 12

17 U.S.C. 102(a)(1) ... 1

17 U.S.C. 102(b) ... 2, 7, 8, 12, 15, 16

17 U.S.C. 106(1) ... 3

17 U.S.C. 106(2) ... 3, 18

Statutes, regulation, and rule—Continued: Page

Miscellaneous:

In the Supreme Court of the United States

No. 18-956

GOOGLE LLC, 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 "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 creativity." *Feist Publ'ns, Inc.* v. *Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991). The copyright

in an original literary work encompasses both its literal aspects (*i.e.*, the actual text) and its original non-literal aspects, such as the plot of a novel. See, *e.g.*, *Twin Peaks Prods., Inc.* v. *Publications Int'l, Ltd.*, 996 F.2d 1366, 1372-1373 (2d Cir. 1993).

The Copyright Act limits the scope of the rights conferred by a copyright. As particularly relevant here, Section 102(b) states 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). For example, a copyright for a book that explains how to perform a new surgical method would bar others from copying the book without authorization, but not from performing the surgical method. Section 102(b) codifies the longstanding common-law principle known as the "idea/expression dichotomy." *Golan* v. *Holder*, 565 U.S. 302, 328 (2012); see H.R. Rep. No. 1476, 94th Cong., 2d Sess. 56-57 (1976) (1976 House Report).

Other common-law doctrines limit the copyrightability of certain expressive works. 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 "standard, stock, or common to a topic," like cowboys and shootouts in stories of the American West, are not copyrightable. *Mitel, Inc.* v. *Iqtel, Inc.*, 124 F.3d 1366, 1374 (10th Cir. 1997); see *Zalewski*, 754 F.3d at 102.

b. A valid copyright gives the owner certain "exclusive rights," including the rights "to reproduce the copyrighted work" and "to prepare derivative works based upon the copyrighted work." 17 U.S.C. 106(1) and (2). But those rights are subject to limitations, including 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 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": (1) "the purpose and character of the use," (2) "the nature of the copyrighted work," (3) "the amount and substantiality of the portion used in relation to the copyrighted work as a whole," and (4) "the effect of the use upon the potential market for or value of the copyrighted work." 17 U.S.C. 107(1)-(4).

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 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 "firmly established" and undisputed in this case that computer code can be copyrightable as a "literary work[]." 1 Melville B. Nimmer & David Nimmer, *Nimmer on Copyright* § 2A.10[B] (2019). 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." 17 U.S.C. 101. And various 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. 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. Pet. App. 4a.

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 tasks such as displaying text on the screen or playing a sound. In Java, these subprograms are called "methods." Pet. App. 125a. Respondent's predecessor-in-interest created a "Standard Library" of thousands of prewritten methods, which have been organized into "classes," which in turn are organized into "packages." *Ibid.* The version of the Standard Library at issue in this case, Java SE, includes 166 packages, comprising more than 600 classes and more than 6000 methods. See *id.* at 126a. Although a Java programmer can 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 in the Java language itself, it owns a copyright in the Java Standard Library. Pet. App. 127a. Respondent makes the Java Standard Library available to computer programmers under three copyright licenses, including royalty-free options. *Ibid.* For a commercial license, respondent requires the licensee to ensure that programs it creates using the Java Standard Library can be run using the Java platform. *Id.* at 127a-128a.

b. In general, to create a new Java method, a programmer must write 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 refer to the first type of code as "declaring code" and to the second as "implementing code." Pet. 5.

An example drawn from the district court's opinion illustrates the distinction. See Pet. App. 224a-225a. 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*, for an integer); and the type and order of the input data it accepts (integer *x* and integer *y*). (Additional declaring code, not reproduced above, would place the method into a class and classes 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 or "call" the method by typing a command consisting of the name of the method 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, a programmer can use the prewritten methods in respondent's Java Standard Library without seeing or understanding 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. The Android platform uses the Java programming language. As a result of petitioner's design choices, however, applications written for Android do not run on the Java platform, and vice versa. See Pet. App. 46a n.11, 130a, 172a.

Like the Java platform, petitioner's Android platform contains a collection of prewritten methods organized into classes and packages. 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. Pet. App. 129a. The copied packages contained the Java methods and classes that petitioner viewed as most useful for developing smartphone applications. See Pet. 25. Petitioner asserts that it copied the declaring code so that programmers familiar with the Java platform would be able to program for the Android platform without learning new commands for invoking commonly used methods. See Pet. 25-26.

Petitioner copied 11,500 lines of respondent's copyrighted code. Pet. App. 7a. In doing so, petitioner also copied the complex architecture of the 37 packages at issue, including the names and specifications of the thousands of methods and classes in those packages and their hierarchical and interdependent relationships to each other. See *id.* at 134a.

3. In August 2010, respondent sued petitioner in the Northern District of California, alleging that petitioner had infringed respondent's copyright in the Java Standard Library and had also infringed related patents. Pet. App. 2a, 97a n.2. Respondent's copyright-infringement claims ultimately proceeded on two theories: (i) literal, verbatim copying of the declaring code; and (ii) nonliteral copying of the "structure, sequence, and organization" (SSO) of the Java Standard Library, which the declaring code establishes and reflects. See *id.* at 2a, 132a. The case proceeded to trial, and a jury found infringement but hung on fair use. *Id.* at 122a.

The district court set aside the infringement verdict on the ground that respondent did not possess a valid copyright in the copied material. Pet. App. 212a-272a. The court held 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 267a. The court also held that the merger doctrine rendered the declaring code uncopyrightable. *Id.* at 264a.

The Federal Circuit, which had appellate jurisdiction because of petitioner's since-abandoned patent claims, affirmed in part and reversed in part. Pet. App. 121a-192a; see 28 U.S.C. 1295(a)(1). Applying Ninth Circuit precedent (see Pet. App. 134a), the court of appeals held that the declaring code and the SSO were both "entitled to copyright protection," despite their functional character. *Id.* at 123a. The court explained that Section 102(b) "'restate[s] * * * the basic dichotomy between expression and idea,'" but does not "take[] * * * away" copyright protection from works that otherwise satisfy the criteria in Section 102(a). *Id.* at 141a (quoting *Feist Publ'ns*, 499 U.S. at 356). The court also held that neither the merger nor the *scènes-à-faire* doctrine precluded copyright protection, noting (among other things) that when respondent's predecessor-in-interest created the Java Standard Library, it "had 'unlimited options as to the selection and arrangement'" of the code that petitioner ultimately copied. *Id.* at 150a (citation omitted); see *id.* at 156a-157a. Finally, the court remanded for a new trial on fair use. *Id.* at 191a.

Petitioner sought this Court's review on the question "[w]hether copyright protection extends to all elements of an original work of computer software * * * that an author could have written in more than one way." 14-410 Pet. i. This Court invited the Solicitor General to file a brief expressing the views of the United States, 135 S. Ct. 1021, and the United States recommended that the petition be denied, see 14-410 U.S. Br. 10-11. This Court denied the petition. 135 S. Ct. 2887.

4. On remand, a jury found that petitioner's copying was fair use, and the district court denied respondent's

post-trial motions for judgment as a matter of law. Pet. App. 9a; see *id.* at 57a, 92a-120a.

A unanimous panel of the Federal Circuit reversed. Pet. App. 1a-55a. Again applying Ninth Circuit precedent (see *id.* at 12a), the court of appeals found that the first and fourth statutory fair-use factors—the "purpose and character of the use" and its "effect * * * upon the potential market for or value of the copyrighted work," 17 U.S.C. 107(1) and (4)—both "weigh[ed] heavily against a finding of fair use." Pet. App. 53a. The court determined that petitioner's copying was "overwhelmingly commercial," *id.* at 28a, and that petitioner had not made "transformative" use of the copied material, *id.* at 31a. In the court's view, the declaring code was used to "perform the same functions in Android and Java," *ibid.*, and petitioner had not changed the copied code's expressive content or message, *id.* at 33a-35a. With respect to the effect of the copying, the court noted "'overwhelming'" record evidence that petitioner's copying had inflicted "actual and potential harm" on the market for respondent's work, including by enabling one of respondent's customers to use the existence of petitioner's Android platform as leverage "to negotiate a steep discount" for continuing to license the Java platform. *Id.* at 50a-51a (citation omitted).

By contrast, the court of appeals viewed the "nature of the copyrighted work," 17 U.S.C. 107(2), as supporting the jury's fair-use finding. The court explained that, although writing the declaring code and the SSO had "involved some level of creativity," a reasonable jury "could have concluded that functional considerations were both substantial and important." Pet. App. 42a. The court viewed the remaining fair-use factor—the "amount and substantiality of the portion used in relation to the copyrighted work as a whole," 17 U.S.C. 107(3)—as "at best[] neutral." Pet. App. 47a; see *id.* at 45a (noting that petitioner had "copied 11,500 lines of code," and that "only 170 lines of code were necessary to write in the Java language").

"Weighing these factors together," the court of appeals concluded that petitioner's "use of the declaring code and SSO * * * was not fair as a matter of law." Pet. App. 53a. The court emphasized, however, that its decision rested on the specific "facts relating to the copying at issue here" and "this particular code." *Id.* at 54a.

DISCUSSION

Petitioner's first question presented is essentially identical to the question on which this Court previously denied certiorari. The United States continues to believe that the court of appeals' holding as to copyrightability is correct, and that the question does not warrant this Court's review. See 14-410 U.S. Br. 19-23.

Petitioner's second question presented (Pet. I) seeks review of the court of appeals' determination that respondent was entitled to judgment as a matter of law on the issue of fair use. Although the question whether to grant judgment as a matter of law on this record is not free from doubt, the government agrees with that holding as well. Petitioner copied 11,500 lines of computer code verbatim, as well as the complex structure and organization inherent in that code, in order to help its competing commercial product. The record demonstrates, moreover, that petitioner's unauthorized copying harmed the market for respondent's Java platform.

In any event, the court of appeals' holding on fair use was limited to the particular facts of this case; those atypical facts make this an unsuitable vehicle for addressing the application of copyright law to software generally; and no other considerations counsel in favor of granting review. The petition for a writ of certiorari therefore should be denied.

I. THE COPYRIGHTABILITY QUESTION DOES NOT WARRANT REVIEW

A. The Court Of Appeals Correctly Held That Neither Section 102(b) Nor The Merger Doctrine Forecloses Copyright Protection

1. Various Copyright Act provisions make clear that a person may own a copyright in a "computer program." 17 U.S.C. 101 (defining "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"); see, *e.g.*, 17 U.S.C. 117 (entitled "Limitations on exclusive rights: Computer programs") (emphasis omitted). A computer program is "expressed in words, numbers, or other verbal or numerical symbols or indicia" and therefore fits comfortably within the Act's definition of "[l]iterary works." 17 U.S.C. 101; see 1976 House Report 54; see also 14-410 U.S. Br. 11-12.

To be copyrightable, any particular computer code must meet the basic requirements of copyright law, including originality, 17 U.S.C. 102(a). But petitioner does not dispute that both the 11,500 lines of declaring code and the structure, sequence, and organization (SSO) of the Java Standard Library meet the originality requirement, see Pet. App. 141a—*i.e.*, that they possess the requisite "minimal degree of creativity," *Feist Publ'ns, Inc.* v. *Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991).

2. Petitioner primarily contends (Pet. 16) that the declaring code is an uncopyrightable "method of operation," 17 U.S.C. 102(b), for accessing prewritten functions implemented in other code that petitioner did not copy. The court of appeals correctly rejected that argument. Pet. App. 165a-166a.

Section 102(b) codifies the "idea/expression dichotomy," *Golan* v. *Holder*, 565 U.S. 302, 328 (2012), under which a copyright in an "original work[] of authorship," 17 U.S.C. 102(a), covers only the expressive work itself—not the underlying ideas or methods of operation "described, explained, illustrated, or embodied" in the work, 17 U.S.C. 102(b). Although there is a sense in which all computer code could be described as a method of operating a computer, the Copyright Act as a whole makes clear that computer programs can be protected by copyright, refuting any suggestion that the functional character of computer code suffices to bring it within Section 102(b). See 14-410 U.S. Br. 13-14.

Petitioner's effort to distinguish the declaring code at issue here from other copyright-eligible computer code is unavailing. Petitioner asserts (Pet. 16) that, 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. Both are necessary components of a Java or Android method. And neither is what a programmer physically types when invoking a method.

Petitioner observes that "the actual processes or methods embodied in [a computer] program are not within the scope of the copyright law." Pet. 17 (citation omitted). But respondent is not asserting a copyright in abstract processes or methods, such as a method for selecting the larger of two integers. Without infringing any copyright, petitioner could and did write its own code to implement the same processes or methods. But petitioner also copied 11,500 lines of respondent's declaring code into Android, thereby replicating in its own program respondent's creative expression in the text and architecture of the Java Standard Library.

3. Petitioner contends (Pet. 17-21) that the merger doctrine precludes copyright protection for the declaring code. The court of appeals correctly rejected that argument. Pet. App. 147a-153a.[1]

The merger doctrine reinforces the idea/expression dichotomy by precluding copyright when an idea can be expressed in only a limited number of ways. In that circumstance, the idea and its expression "merge," and the latter is uncopyrightable. *Zalewski* v. *Cicero Builder Dev., Inc.*, 754 F.3d 95, 103 (2d Cir. 2014). That concern is not implicated here, since respondent's predecessor-in-interest "had 'unlimited options as to the selection and arrangement'" of the lines of code that petitioner copied. Pet. App. 150a (citation omitted).

Petitioner suggests (Pet. 20) that the declaring code could be written "only in one way" after respondent's predecessor-in-interest made certain conceptual decisions about the organization of the Java Standard Library. Cf. Pet. Reply Br. 9. Even if that were true, it would show at most that the declaring code is the only way to express the SSO of the Java Standard Library.

[1] Petitioner's earlier petition did not seek review of the court of appeals' merger holding. See 14-410 U.S. Br. 22.

But the SSO is itself a copyrightable aspect of respondent's original work of authorship—not an uncopyrightable idea that respondent is seeking to monopolize.

Petitioner's reliance (Pet. 17-19) on *Baker* v. *Selden*, 101 U.S. 99 (1880), is thus misplaced. *Baker* involved a copyrighted book that explained a system of accounting and included forms for implementing the system. This 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. *Id.* at 101, 104-105, 107. The premise of *Baker* was that the accounting method itself, unlike the SSO here, was not subject to copyright protection. And nothing about the Java language or the technical constraints of mobile devices compelled petitioner's large-scale copying. "[O]nly 170 lines of code" are "necessary to write in the Java language," Pet. App. 45a, and "Microsoft and Apple developed mobile operating systems from scratch, using their own array of software packages," *id.* at 149a n.5.

Petitioner also argues (Pet. 20-21) that the court of appeals erred by focusing on the choices available to respondent ex ante when it created Java, rather than on the choices available to petitioner when it sought to devise a way for programmers familiar with "industry-standard Java shorthand commands" (Pet. 27) to use those same commands in Android. Petitioner's approach would treat the current popularity of respondent's work among developers as retroactively divesting the work of copyright protection. The court below correctly rejected that approach, finding it "well-established that copyrightability" should be "evaluated at the time of creation." Pet. App. 151a. Petitioner's contrary view 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.

Finally, copying the complete text of the declaring code for 37 packages—comprising more than 11,500 lines of code and reflecting the complex SSO of the Java Standard Library—constituted much more than merely replicating uncopyrightable names (Pet. 20; Pet. Reply Br. 9). Although words and short phrases are not copyrightable on their own, 37 C.F.R. 202.1(a), determining copyrightability requires an evaluation of the work as a whole. See, *e.g.*, *Roth Greeting Cards* v. *United Card Co.*, 429 F.2d 1106, 1109 (9th Cir. 1970). And the decision below does not foreclose the possibility that other declaring code may be uncopyrightable under the merger doctrine—or under Section 102(b), the *scènes-à-faire* doctrine, or other limits on copyrightability. See 14-410 U.S. Br. 14. The court of appeals merely held, correctly, that the particular code at issue here is entitled to copyright protection.

B. Further Review Is Not Warranted

In arguing that the copyrightability question warrants this Court's review, petitioner relies principally (Pet. 12-13) on the same two decisions that petitioner previously invoked (14-410 Pet. 13-16) as the basis of a putative circuit conflict on the application of Section 102(b) to computer programs. See *Lotus Dev. Corp.* v. *Borland Int'l, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995), aff'd by an equally divided Court, 516 U.S. 233 (1996), and *Lexmark Int'l, Inc.* v. *Static Control Components, Inc.*, 387 F.3d 522 (6th Cir. 2004).

The Federal Circuit's construction of Section 102(b) in this case does not conflict with those decisions. See 14-410 U.S. Br. 19-22. In *Lotus*, the First Circuit invoked Section 102(b) to find that the arrangement of menu commands presented to a software user was an uncopyrightable "'method of operation'" for the software at issue. 49 F.3d at 815-818. The case did not address the copyrightability of computer code, and the First Circuit has subsequently acknowledged, consistent with the decision below, that Section 102(b) codifies the idea/expression dichotomy. See *Situation Mgmt. Sys.* v. *ASP Consulting LLC*, 560 F.3d 53, 61 (2009). In *Lexmark*, the Sixth Circuit held that a short computer program that operated as a technological lock to bar the re-use of certain printer ink cartridges was likely uncopyrightable where the features of the program were largely dictated by external constraints, such as technical requirements. 387 F.3d at 529-530, 535-536. No similar constraints compelled petitioner to copy 11,500 lines of respondent's code.

Relying again on *Lexmark*, petitioner also briefly reprises its prior argument (Pet. 14-15; see 14-410 Pet. 19) that the circuits are divided over aspects of the merger doctrine. But the Federal Circuit correctly held that merger principles do not apply here because respondent had ample options in creating its declaring code. Pet. App. 150a-152a. Petitioner does not contend that any other circuit would have reached a different result on these facts. And although petitioner asserts (Pet. 15) that courts are divided on "the role of interoperability in the merger doctrine," this case does not implicate interoperability issues in the relevant sense because petitioner did not make Android interoperable with the Java platform. See p. 6, *supra*.

II. THE FAIR-USE QUESTION DOES NOT WARRANT REVIEW

A. The Court Of Appeals Correctly Held That No Reasonable Jury Could Find Fair Use On This Record

1. The doctrine of fair use limits the exclusive rights a copyright otherwise confers. The doctrine permits courts to consider whether "rigid application of the copyright statute" in a particular case "would stifle the very creativity which that law is designed to foster." *Campbell* v. *Acuff-Rose Music, Inc.*, 510 U.S. 569, 577 (1994) (citation omitted). Section 107 identifies four non-exclusive factors that courts "shall" consider in assessing whether a particular use is fair: (1) "the purpose and character of the use, including whether such use is of a commercial nature"; (2) "the nature of the copyrighted work"; (3) "the amount and substantiality of the portion used in relation to the copyrighted work as a whole"; and (4) "the effect of the use upon the potential market for or value of the copyrighted work." 17 U.S.C. 107(1)-(4). A court must evaluate all four factors in light of "the goal of copyright, to promote science and the arts." *Campbell*, 510 U.S. at 579.

The first and fourth factors are particularly relevant here. Under the first factor, a court examines whether the defendant's use is commercial, *Campbell*, 510 U.S. at 578, and whether it "adds something new, with a further purpose or different character," *id.* at 579. Such a "transformative" use is more likely to be deemed fair use because it furthers the goals of copyright by fostering creative expression. *Ibid.* As to the fourth factor, because copying that usurps the original work discourages authors from investing the effort that creative expression entails, a defendant "would have difficulty carrying the burden of demonstrating fair use without favorable evidence about relevant markets." *Id.* at 590. Relevant markets include "the market for derivative works," *Harper & Row, Publishers, Inc.* v. *Nation Enters.*, 471 U.S. 539, 568 (1985), which copyright holders enjoy the exclusive right to create and license. 17 U.S.C. 106(2).

2. The court of appeals concluded that, on this record, the first and fourth Section 107 factors weigh so heavily against fair use that the second and third cannot tip the balance in petitioner's favor. Pet. App. 53a. Although a jury verdict should not be lightly set aside, the court's decision was correct. Petitioner's criticisms (Pet. 24-29) of the court's approach lack merit.

a. On the first factor, petitioner contends (Pet. 24) that the court of appeals improperly "fixat[ed]" on the code that petitioner copied, rather than examining the Android platform as a whole. But the court recognized the relevance of "what [declaring code] does in Java and in Android, how the audience of computer developers perceives it, how much [petitioner] took and added, [and] what the added code does." Pet. App. 33a. And of course, "no plagiarist can excuse the wrong by showing how much of his work he did not pirate." *Harper & Row*, 471 U.S. at 565 (citation omitted).

The court of appeals also correctly held that petitioner had not transformed respondent's code by integrating a verbatim copy into the Android platform. Pet. App. 35a-37a. Petitioner used the declaring code for the same purpose for which it was created, without any changes to the expression or to the meaning or message of the packages. Cf. *TCA Television Corp.* v. *McCollum*, 839 F.3d 168, 181-182 (2d Cir. 2016) (not transformative use to copy a comedic routine, without altering its meaning, into the new context of a dramatic play), cert. denied, 137 S. Ct. 2175 (2017). Petitioner also did not transform the copied work by using it in the "new environment" (Pet. 26) of the Android platform, just as a copier does not ordinarily give a copyrighted poem a "further purpose or different character" by including it in his own book of poetry, *Campbell*, 510 U.S. at 579; see also, *e.g.*, *Disney Enters., Inc.* v. *VidAngel, Inc.*, 869 F.3d 848, 861 (9th Cir. 2017).

Computer code *can* be used in transformative ways, such as by excerpting it in a textbook to illustrate a coding technique. And lower courts have wrestled with issues, not presented here, about whether making temporary copies of existing code to "reverse engineer" a system, in order to create compatible works that do not incorporate the pre-existing code, constitutes fair use. See, *e.g.*, *Sony Computer Entm't, Inc.* v. *Connectix Corp.*, 203 F.3d 596, 603-605 (9th Cir. 2000); *Sega Enters. Ltd.* v. *Accolade, Inc.*, 977 F.2d 1510, 1525-1527 (9th Cir. 1993); *Atari Games Corp.* v. *Nintendo of Am. Inc.*, 975 F.2d 832, 844-845 (Fed. Cir. 1992); see also U.S. Copyright Office, *Software-Enabled Consumer Products* 54-59 (2016).[2] Courts have generally found that copying code to discern how an existing product works, in order to ensure that a new (non-infringing) product is interoperable with the existing product, is a transformative use. But here, petitioner took lines of code from a rival software platform to make a competing platform that is not interoperable with the Java platform. See p. 6, *supra*.

[2] https://go.usa.gov/xVNyD.

Petitioner argues (Pet. 25-26) that its copying promoted "interoperability" by allowing developers for Android to draw on their preexisting knowledge of the Java platform. Petitioner identifies no court that has accepted that expansive conception of interoperability, which is inconsistent with the Copyright Act's definition of the term, see 17 U.S.C. 1201(f)(4) (defining "'interoperability'" for purposes of Section 1201(f) to mean "the ability of computer programs to exchange information, and of such programs mutually to use the information which has been exchanged"). Petitioner's idiosyncratic approach would seem to allow any copyist to carve out the most popular parts of a pre-existing work, on the ground that familiar content is likely to make the second work more commercially appealing to admirers of the first. That result would be antithetical to the purposes of copyright.

b. On the fourth factor, the trial record contained "'overwhelming'" evidence that petitioner's copying harmed the market for Java, including evidence that one of respondent's customers used the existence of the Android platform as leverage "to negotiate a steep discount" for continuing to use the Java platform in the customer's tablets. Pet. App. 50a-51a (citation omitted).

More broadly, the fair-use doctrine does not permit copying valuable parts of a work to attract fans to a competing commercial product. Copying "to get attention or to avoid the drudgery in working up something fresh" disserves copyright's goals. *Campbell*, 510 U.S. at 580. And petitioner's copying was not necessary to foster innovation in this context, as demonstrated by the commercial success of other, non-infringing mobile platforms like Apple's.

c. Petitioner contends (Pet. 23-24) that the court of appeals' fair-use analysis did not take account of the "functional nature" of computer code. But the Copyright Act identifies the "nature of the copyrighted work" as one of the factors that bears on every fair-use determination. 17 U.S.C. 107(2); see *Campbell*, 510 U.S. 586. The functional nature of software can often be an important consideration under that factor.

In this case, the court of appeals held that the second fair-use factor favored petitioner because the jury could reasonably have concluded that "functional considerations were both substantial and important" in creating the declaring code and SSO. Pet. App. 42a. As discussed above, the analysis used to determine whether a particular use is transformative may be different for computer programs than for traditional literary works. But the court of appeals did not suggest otherwise, and its "overall approach" (Pet. 23) was properly tailored to the specific facts of this case. To the extent petitioner believes that copyright law should accommodate "the development and use of * * * technologies" in a different manner, petitioner is "free to seek action from Congress." *American Broad. Cos.* v. *Aereo, Inc.*, 573 U.S. 431, 451 (2014).

B. The Court Of Appeals' Fact-Bound Decision On Fair Use Does Not Warrant Further Review

Petitioner identifies no sound basis for further review of the fair-use issue. The court of appeals limited its fair-use holding to the facts of this case and disclaimed any broader fair-use rule governing other "action[s] involving the copying of computer code." Pet. App. 53a-54a. And petitioner does not contend that the court of appeals' fair-use decision conflicts with the decision of any other court. Cf. Sup. Ct. R. 10(a).

Many of petitioner's amici raise concerns about the potential impact of the fair-use decision on industry practices. See, *e.g.*, Microsoft Amicus Br. 3-5. But the court of appeals simply endorsed the unremarkable proposition that wholesale copying of thousands of lines of copyrighted code into a competing commercial product for the purpose of attracting developers familiar with the copyright owner's work, while causing actual commercial harm to the copyright owner, is not fair use. And because of the intensely fact-bound quality of the decision below, the case would be an unsuitable vehicle for this Court to establish any broad principles of fair-use jurisprudence. Even a decision reversing the Federal Circuit might establish nothing more than that the jury's finding of fair use was reasonable on this record.[3]

Finally, additional idiosyncratic features of this case continue to counsel against further review. See 14-410 U.S. Br. 22-23. The decision below is of limited precedential value because it does not bind either future Ninth Circuit panels or future Federal Circuit panels in appeals from district courts outside the Ninth Circuit. And 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 from a "platform" of programming tools designed to assist programmers in

[3] This case is thus unlike the "seminal" decisions petitioner invokes (Reply Br. 2), in which the Court addressed nationally significant questions of copyright law. See *Campbell*, 510 U.S. at 571-572 (whether a commercial parody can be fair use); *Harper & Row*, 471 U.S. at 542 (whether the publication of excerpts of President Ford's previously unpublished memoir was fair use); *Sony Corp. of Am.* v. *Universal City Studios, Inc.*, 464 U.S. 417, 447-456 (1984) (whether recording a television broadcast on home video for the non-commercial purpose of watching it at a later time is fair use).

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 are familiar with respondent's prewritten methods—that may have little significance in more typical disputes.

CONCLUSION

The petition for a writ of certiorari should be denied.

Respectfully submitted.

NOEL J. FRANCISCO
Solicitor General
JOSEPH H. HUNT
Assistant Attorney General
MALCOLM L. STEWART
Deputy Solicitor General
MATTHEW GUARNIERI
Assistant to the Solicitor General
MARK R. FREEMAN
DANIEL TENNY
SONIA M. CARSON
Attorneys

SEPTEMBER 2019