What the Supreme Court is likely to do in the presently pending case Google v. Oracle

Jennifer Campbell

What the Supreme Court is likely to do in the presently pending case *Google v. Oracle*.

By: Jennifer Campbell

I.   Introduction

Advances in technology, specifically relating to computer science, have evolved rapidly, and it has been difficult for the law to keep up.[1] The pending U.S. Supreme Court case, *Google LLC v. Oracle America, Inc.*, demonstrates some issues that have emerged in copyright law with the rapid growth of technology in computer science.

In 2008, Google released its Android platform, software that Google and other manufacturers use in smartphones and tablets.[2] Since Android's release, Google has produced over $42 billion in revenue solely from the Android platform's advertising.[3] After Oracle America acquired Sun Microsystems in 2010, it filed suit against Google, alleging that Google's Android platform infringed on Oracle's Java SE platform.[4] The litigation between Oracle America and Google has been ongoing for a decade, and now the litigation is pending before the U.S. Supreme Court.[5] The dispute stems from Google's use of computer programs, known as application programming interfaces, or API packages, written in the Java language to develop the Android platform.[6] Software interfaces, like the APIs at issue, are "functional systems or methods of operating a computer program or platform" and "merely describe what functional tasks a computer program will perform without specifying how it does so."[7] Software interfaces are organized by what courts commonly refer to as the "structure, sequence, and organization."[8] The Java platform is an innovative platform that was the first to allow computer programmers to write one program that could run

---

[1] *See* Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693, 696 (2d Cir. 1992).

[2] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1187-88 (Fed. Cir. 2018).

[3] *Id.* at 1187.

[4] *See* Petition for Writ of Certiorari at 7, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2019 WL 338902.

[5] *See* Oracle America, Inc. v. Google, Inc., 872 F. Supp. 2d 974, 974 (N.D. Cal. 2012); Petition for Writ of Certiorari, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2019 WL 338902; Google LLC v. Oracle America, Inc., 140 S. Ct. 520 (2019).

[6] Oracle America, Inc. v. Google Inc., 750 F.3d 1339, 1347, 1351 (Fed. Cir. 2014).

[7] Brief Amici Curiae of 83 Computer Scientists in Support of Petitioner at 2-3, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2020 WL 242506.

[8] *Id.* at 6.

on multiple operating systems.[9] Before Java, programmers had to rewrite their programs for each different computer type.[10]

The first issue that arose in the litigation between Oracle and Google was whether the declaring code and the overall structure, sequence, and organization of thirty-seven of Oracle's API packages were copyrightable.[11] The U.S. District Court for the Northern District of California found that the API packages at issue were not subject to copyright protection.[12] Thereafter, the U.S. Court of Appeals for the Federal Circuit reversed the lower court's decision concluding that the API packages were indeed subject to copyright protection. The appellate court remanded the case back to the district court to determine whether Google had a fair use defense under the Copyright Act.[13] The remand led to a lengthy jury trial, and the district court entered judgment in favor of Google finding the company's use of the copyrighted API material constituted fair use.[14] On appeal, the Federal Circuit again reversed the district court's judgment, this time finding that Google's use was not fair as a matter of law.[15] Google petitioned the U.S. Supreme Court, and the Supreme Court granted certiorari on November 15, 2019.[16] The two issues presently pending in the Supreme Court are: (1) whether Java's declaring code and the structure, sequence, and organization of the APIs are copyrightable, and (2) whether Google's use of the declaring code and structure, sequence, and organization of the APIs constituted fair use, along with the Supreme Court's *sua sponte* request regarding the appropriate standard of review on the fair use issue.[17]

This article seeks to provide applicable principles, rules, statutes, and relevant case law relating to copyrights. Additionally, this article details the circuit court splits on what is protectable expression in

---

[9] Oracle America, Inc. v. Google Inc., 750 F.3d at 1348.

[10] *Id.*

[11] Oracle America, Inc. v. Google Inc., 750 F.3d at 1347.

[12] Oracle America, Inc. v. Google Inc., 872 F. Supp. 2d at 1002.

[13] Oracle America, Inc. v. Google Inc., 750 F.3d at 1347.

[14] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1186 (Fed. Cir. 2018).

[15] *Id.*

[16] Petition for Writ of Certiorari at I, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2019 WL 338902; Google LLC v. Oracle America, Inc., 140 S. Ct. 520 (2019).

[17] Petition for Writ of Certiorari, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020); Google LLC v. Oracle America, Inc., No. 18-956, 2020 U.S. LEXIS 2621 (May 4, 2020).

2

computer programs. This article further discusses the Federal Circuit's *Oracle America, Inc. v. Google Inc*. and *Oracle America Inc. v. Google LLC* opinions, analyzing what the Federal Circuit got right in its judgment and what it got wrong. Finally, this article considers the likely approach and conclusions of the Supreme Court's upcoming, long-awaited opinion.

II.   Background

Copyrightability of Software Interfaces

Concepts and Principles of Copyright Law

The drafters of the United States Constitution gave Congress the power to enact laws to protect the "exclusive rights" of authors in their original works.[18] Pursuant to the Copyright Act of 1976, original literary works of authorship, among others, receive copyright protection.[19] When Congress enacted the Copyright Act of 1976, the legislative history explains that Congress included computer programs to the category of literary works, but only to the extent that the computer program incorporates the author's original expression of the idea and not the idea itself.[20]

Copyright law intends to protect an author's creative, original expression of an idea but does not protect the underlying idea itself.[21] As early as 1879, the Supreme Court conveyed the idea-expression dichotomy principle in *Baker v. Selden*.[22] When enacting the Copyright Act of 1976, Congress codified the idea-expression dichotomy, stating that copyright protection in an original work of authorship 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."[23] The

---

[18] *See* U.S. Const., art. I, §8.
[19] *See* 17 U.S.C. §102(a).
[20] H.R. Rep. No. 94-1476, at 54 (1976), *as reprinted in* 1976 U.S.C.C.A.N. 5659, 5667.
[21] *Id.* at 56-57.
[22] 101 U.S. 99, 102-03, 107 (1879) (explaining that the copyright protects the author's expression of how the accounting system is used but it does not protect others from using the accounting system as the author has explained it or from creating a book explaining the same accounting system).
[23] 17 U.S.C. §102(b).

3

legislative history of the Copyright Act of 1976 reflects concerns relating to the extent of copyright protection in computer programs. Congress attempted to address these concerns by explaining that a programmer's expression is the copyrightable element of the computer program, and the "actual process or methods embodied in the program" are not copyrightable.[24]

Sometimes, an author's original expression is not divisible from the idea. Pursuant to the merger doctrine, an idea and the idea's expression merge when there are one or very few ways of expressing the idea.[25] In a Fifth Circuit Court of Appeals case, a gas pipeline company brought a copyright infringement suit against a competitor, alleging that the competitor copied its proposed route of a natural gas pipeline as depicted on a map that the company created.[26] The appellate court agreed with the district court's holding that "the idea of the location of the pipeline" and its expression of that pipeline on a map are inseparable and that the map expressed "the only effective way the idea of the location of the pipeline."[27] Thus, the route of the proposed pipeline was not copyrightable because the idea and expression merged.[28] To determine whether an idea and expression have merged, courts must look at whether there are multiple ways to express the idea in question.[29] If a court determines that the idea and the expression of the idea are indivisible, then the merger doctrine is applicable, and the expression of that idea is not protectable by copyright.[30]

Another important concept in copyright law relates to the copyrightability of compilations.[31] An author's work must be original to receive copyright protection.[32] Facts are not original, and therefore, not copyrightable.[33] Although factual compilations may include some originality, that originality does not

---

[24] H.R. Rep. No. 94-1476, at 57 (1976), *as reprinted in* 1976 U.S.C.C.A.N. 5659, 5670.

[25] Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983).

[26] Kern River Gas Transmission Co. v. Coastal Corp., 899 F.2d 1458, 1461 (5th Cir. 1990).

[27] *Id.* at 1363-64.

[28] *Id.*

[29] Mason v. Montgomery Data, Inc., 967 F.2d 135, 138-39 (5th Cir. 1992).

[30] *Id.* at 139.

[31] The Copyright Act defines a compilation as "a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship." 17 U.S.C. §101.

[32] Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340, 345 (1991).

[33] *Id.* at 347.

4

extend to the facts, but rather the author's selection and arrangement of those facts, so long as the author independently chose the selection and arrangement of the facts and the selection and arrangement involve a "minimal degree of creativity."[34] Only the author's selection or arrangement is considered original and protectable by copyright law.[35] In *Feist Publications, Inc. v. Rural Telephone Service Co.*, a telephone utility company brought a copyright infringement action against a publisher of telephone directories, alleging that the publisher copied the utility company's white page listings from its telephone directory.[36] The utility company asserted that the sweat of the brow doctrine[37] applied, which meant that, instead of the publisher copying the listings from the utility company's telephone directory, the publisher was required to go door-to-door to obtain that information.[38] The Supreme Court rejected the sweat of the brow approach to copyright protection, stating that "originality, not 'sweat of the brow,' is the touchstone of copyright protection in directories and other fact-based works."[39] Thereafter, the Supreme Court concluded that the information provided in the telephone utility company's telephone directory was uncopyrightable facts.[40] Finally, the Court determined that the telephone directory lacked originality in the coordination and arrangement of those facts, that the arrangement of names in alphabetical order lacked any degree of creativity, and that such an arrangement is commonplace in telephone directories.[41]

U.S. Circuit Court Approaches to Determine What is Protectable Expression in Computer Programs

The United States Courts of Appeals do not uniformly agree on how to determine what is protectable expression in computer programs. The U.S. Courts of Appeals for the First and Sixth Circuits' view is that 17 U.S.C. §102(b)[42] precludes copyright protection of expression since the

---

[34] *Id.* at 348.

[35] *Id.* at 350.

[36] *Id.* at 344.

[37] The underlying notion of the sweat of the brow doctrine "was that copyright was a reward for the hard work that went into compiling facts." *Id.* at 352.

[38] *Id.* at 344.

[39] *Id.* at 359-60.

[40] *Id.* at 361.

[41] *Id.* at 362-63.

[42] Section 102(b) states, "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. §102(b).

5

expression is contained in forms that are not copyrightable, such as procedures, processes, or methods of operation.[43] Meanwhile, the U.S. Court of Appeals for the Third Circuit's view is that 17 U.S.C. §102(b) does not preclude copyright protection so long as there are multiple ways to express the idea.[44] Finally, the U.S. Court of Appeals for the Second Circuit takes an entirely different approach by applying an abstraction-filtration-comparison test.[45]

In the U.S. Court of Appeals for the First Circuit's case, *Lotus Development Corp. v. Borland International, Inc.*, the parties disputed whether Lotus's menu command hierarchy was copyrightable.[46] The lower court initially took a similar approach to the Third Circuit's approach, concluding that because the Lotus developers had a variety of choices on how to structure the menu command hierarchy and a choice of words to use in the menu, consequently the menu command hierarchy constituted copyrightable expression.[47] However, the First Circuit focused not on whether the copied material was expression, but whether it was protectable under 17 U.S.C. §102(b).[48] In doing so, the court rejected the Second Circuit's abstraction-filtration-comparison test, concluding that, since the test only applies to nonliteral copying, the test would be of little assistance with the instant issue of literal copying.[49] The court concluded that the menu command hierarchy was a method of operation and, thus, uncopyrightable.[50] After the First Circuit's ruling, Lotus petitioned for certiorari to the U.S. Supreme Court. The High Court granted certiorari, and an equally divided Court affirmed the First Circuit's judgment.[51]

The U.S. Court of Appeals for the Sixth Circuit came to a similar conclusion as the First Circuit in *Lexmark International, Inc. v. Static Control Components, Inc*. Similar to *Lotus*, *Lexmark* dealt with

---

[43] *See* Lotus Development Corp. v. Borland International, Inc., 49 F.3d 807 (1st Cir. 1995) *aff'd*, 516 U.S. 233 (1996); Lexmark International, Inc. v. Static Control Components, Inc., 387 F.3d 522 (6th Cir. 2004).
[44] *See* Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986).
[45] *See* Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992).
[46] 49 F.3d 807, 809 (1st Cir. 1995) *aff'd*, 516 U.S. 233 (1996).
[47] *Id.* at 811.
[48] *See Id.* at 815.
[49] *Id.* at 814-15.
[50] *Id.* at 815.
[51] *See* Lotus Development Corp. v. Borland International, Inc., 516 U.S. 233 (1996).

6

the literal copying of aspects of a computer program, specifically an exact copy of Lexmark's Toner Loading Program.[52] Also similar to *Lotus*, the lower court in *Lexmark* deemed the program copyrightable because a programmer could write it in various ways.[53] The Sixth Circuit discussed the two-prong test for analyzing a copyright infringement claim.[54] The first prong, which looks at the ownership of a valid copyright, "tests the originality and non-functionality of the work."[55] In the second prong, a court asks whether the defendant copied protectable elements of the work; this prong tests whether copying occurred and if so, whether the copied portions were protectable under copyright.[56] The appellate court determined that the lower court erred in its reasoning under the second prong and explained that to determine if originality exists, a "court should ask whether the ideas, methods of operation and facts of the program could have been expressed in any form other than that chosen by the programmer, taking into consideration the functionality, compatibility and efficiency demanded of the program."[57] The court's determination implied that expression falling under any of the categories in 17 U.S.C. §102(b) precludes copyright protection.[58]

The U.S. Court of Appeals for the Third Circuit takes a contrary approach from the First and Sixth Circuits. In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, a dispute arose about the similarities between the structure and organization of two programs used in dental laboratories.[59] However, the code in the programs was not the same.[60] The lower court determined that the defendant did not create his program independently, that he had access to the plaintiff's program, and that the two programs were substantially similar to one another, concluding that the defendant infringed on the plaintiff's copyright.[61] The court set out to determine whether the structure of a program was

---

[52] 387 F.3d 522, 530-31 (6th Cir. 2004).
[53] *Id.* at 531.
[54] *Id.* at 534.
[55] *Id.*
[56] *Id.*
[57] *Id.* at 539.
[58] *See Id.*
[59] 797 F.2d 1222, 1228-29 (3d Cir. 1986).
[60] *Id.* at 1233.
[61] *Id.*

7

copyrightable.[62] In answering that question, the court used the *Baker v. Selden* reasoning to conclude that "the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea."[63] The court further explained its conclusion by stating, "[w]here there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea."[64] The court proceeded to apply the rule it created, which the court stated has its "greatest force in the analysis of utilitarian or 'functional' works," to conclude that the structure of the program was part of the programmer's expression and was not the programmer's idea of the program.[65]

The U.S. Court of Appeals for the Second Circuit takes an entirely different approach from the First, Sixth, and Third Circuits. In *Computer Associates International, Inc. v. Altai, Inc.*, the plaintiff initially sued the defendant for literal copying of the code of the plaintiff's program, ADAPTER, in the defendant's program OSCAR 3.4.[66] The defendant did not learn of the code copying until receiving the plaintiff's complaint, but one of the defendant's employees, who was a former employee of the plaintiff, confirmed the copying as true.[67] The defendant then rewrote its OSCAR 3.4 program to exclude any portion of the code copied from ADAPTER.[68] The rewritten OSCAR 3.4, named OSCAR 3.5, then sparked the plaintiff to file a separate, new copyright infringement claim arising from non-literal elements of ADAPTER that the OSCAR 3.5 program purportedly took.[69] The lower court found that OSCAR 3.5's structure, including its non-literal elements, was not substantially similar to ADAPTER.[70] The plaintiff appealed the decision, claiming that the lower court failed to find that copyright protects the expression in non-literal elements of a computer program.[71] The appellate court began with a lengthy analysis of

---

[62] *Id.* at 1234.
[63] *Id.* at 1236.
[64] *Id.*
[65] *Id.* at 1238-39.
[66] 982 F.2d 693, 700 (2d Cir. 1992).
[67] *Id.*
[68] *Id.*
[69] *Id.* at 701.
[70] *Id.* at 696-97.
[71] *Id.* at 702.

8

copyright law, including *Baker v. Selden.*[72] From its analysis, the court concluded that "elements of a

computer program that are necessarily incidental to its function are . . . unprotectable."[73] From there, the

court formulated its abstraction-filtration-comparison test to determine how to separate an idea in a

computer program from its expression.[74] In the first step of the test, "abstraction," a court is to dissect the

plaintiff's copyrighted work's structure and "isolate each level of abstraction contained within it," starting

with the code and ending with "an articulation of the program's ultimate function."[75] The second step,

"filtration," separates the protectable expression from any non-protectable elements by examining the

levels of abstraction "to determine whether their particular inclusion at that level was 'idea' or was

dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors

external to the program itself; or taken from the public domain and hence is nonprotectable expression."[76]

The court elaborated that oftentimes, a programmer's choice in design is made by external factors such

as:

> (1) the mechanical specification of the computer on which a particular program is
> intended to run; (2) compatibility requirements of other programs with which a program
> is designed to operate in conjunction; (3) computer manufacturers' design standards; (4)
> demands of the industry being served; and (5) widely accepted programming practices
> within the computer industry.[77]

The third and final step in the test is "comparison."[78] In the comparison step, after the court has filtered

out all nonprotectable elements in the program, the court compares the remaining protectable expression

in the plaintiff's copyrighted program to the defendant's alleged infringing program for substantial

similarity.[79]

---

[72] *Id.* at 702-05.
[73] *Id.* at 705.
[74] *Id.* at 706.
[75] *Id.* at 707.
[76] *Id.*
[77] *Id.* at 709-10.
[78] *Id.* at 710.
[79] *Id.* at 710-11.

9

The U.S. Court of Appeals for the Ninth Circuit appears to have endorsed the Second Circuit's abstraction-filtration-comparison test as the proper approach.[80] In the court's fair use analysis, it criticized the Third Circuit's approach in *Whelan* as being "simplistic and overbroad," pointing out that a program is an intricate set of subroutines, which are in themselves programs, and each subroutine has its own idea, making the *Whelan* approach inadequate.[81] The court then analyzed the Second Circuit's approach, concluding that "in light of the essentially utilitarian nature of computer programs, the Second Circuit's approach is an appropriate one."[82]

The Fair Use Doctrine

Although copyright holders have exclusive rights to their original expression, there are limitations to their exclusive rights.[83] One of those limitations is fair use.[84] According to 17 U.S.C. §107, if the use of a copyrighted work is fair, then using the copyrighted work is not infringement. The statute also provides a non-comprehensive list of purposes in which the use could be deemed fair: "for purposes such as criticism, comment, news reporting, teaching . . . , scholarship, or research."[85] The statute also provides courts with four factors that they must include when determining whether the use was fair:

> (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
>
> (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.[86]

---

[80] *See* Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).
[81] *Id.* at 1524-25.
[82] *Id.* at 1525.
[83] *See* 17 U.S.C. §107.
[84] *Id.*
[85] *Id.*
[86] *Id.*

10

As the statute's use of the term "shall include" indicates, and as the legislative history explains, courts can consider other relevant factors besides the four that the statute lists when determining fair use.[87] The legislative history further explains that codifying the defense of fair use to copyright infringement claims was not intended to limit the scope of the judicial doctrine of fair use, which is where the doctrine originated, nor to "change, narrow, or enlarge it in any way."[88]

Factor One: The Purpose and Character of the Use

The first factor in a court's fair use analysis is the purpose and character of the use.[89] In 1984, the U.S. Supreme Court concluded that if the copied material is for a commercial or profit-making purpose, then the use would presumptively be unfair.[90] The following year, the Supreme Court solidified the presumption and again stated that "[t]he fact that a publication was commercial as opposed to nonprofit is a separate factor that tends to weigh against a finding of fair use."[91] The Court elaborated on the presumption by explaining that the commercial purpose should focus on whether the copier will profit from exploiting the copyrighted material without paying the established price instead of the motive being solely for monetary gain.[92] Then, in 1994, the Supreme Court seemed to draw back on their presumption that works of a commercial purpose are presumptively unfair.[93] Furthermore, the Court stated that commercial or nonprofit purpose is only one of the elements in the first-factor inquiry of fair use and that if the factor hinged on that one element, then many of the purposes in the statute would fail as they tend to be done for a profit.[94] The Court's factor one inquiry focused more on whether the copying work was transformative from the original.[95] A transformative work does not supplant the original, but rather "adds something new, with a further purpose or different character, altering the first with new expression,

---

[87] H.R. Rep. No. 102-836, at 1 (1992), *as reprinted in* 1992 U.S.C.C.A.N. 2553, 2553.
[88] H.R. Rep. No. 94-1476, at 66 (1976), *as reprinted in* 1976 U.S.C.C.A.N. 5659, 5680.
[89] 17 U.S.C. §107(1).
[90] Sony Corp. of America v. Universal City Studios, Inc., 464 U.S. 417, 448-49 (1984).
[91] Harper & Row, Publishers, Inc v. Nation Enterprises, 471 U.S. 539, 562 (1985.)
[92] *Id.*
[93] Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 583-84 (1994).
[94] *Id.* at 584.
[95] *Id.* at 578-83.

11

meaning, or message."[96] The Court also noted that finding that a work is not transformative does not always go against a finding of fair use.[97] In the *Campbell* case, the Court found that although a parody has a commercial purpose, the work was transformative, which pulled the first factor more toward a finding of fair use.[98]

Factor Two: The Nature of the Copyrighted Work

The second factor in a fair use analysis is the nature of the copyrighted work.[99] "This factor calls for recognition that some works are closer to the core of intended copyright protection than others, with the consequence that fair use is more difficult to establish when the former works are copied."[100] This means that the more creative an author's original expression, like a fictional novel, the stronger the copyright protection, and the less creative, like a factual compilation, the "thinner" the copyright protection.[101] In the Court's analysis of the second factor in *Harper & Row, Publishers, Inc.*, it appears that instead of looking at the nature of the copyrighted work as a whole, the Court focused on the particular parts of the work copied and weighed the expressive elements taken against the facts taken, in which the Court concluded that the use of the expressive elements went beyond what was necessary to convey the facts.[102]

Factor Three: The Amount and Substantiality of the Portion Used

---

[96] *Id.* at 579.
[97] *Id.*
[98] *Id.* at 578-85, 594.
[99] 17 U.S.C. §107(2).
[100] Campbell, 510 U.S. at 586.
[101] *See* Stewart v. Abend, 495 U.S. 207, 237-38 (1990); Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340, 349 (1991).
[102] Harper & Row, Publishers, Inc v. Nation Enterprises, 471 U.S. 539, 563-63 (1985).

12

The third factor in a fair use analysis is the amount and substantiality of the portion used in relation to the copyrighted work as a whole.[103] The analysis looks at the quantity of the plaintiff's copyrighted work that the defendant's work used in relation to the whole of the plaintiff's copyrighted work.[104] The analysis also looks at the quality of the plaintiff's work that the defendant's work used.[105] The Supreme Court stated that a relevant question to this analysis is whether the defendant copied "a substantial portion" of the defendant's work exactly from the plaintiff's original work.[106] If the defendant did copy a substantial portion from the plaintiff's original work, it could indicate that the defendant's work is more likely to take the place of the use and demand for the plaintiff's original work.[107] However, a court could still find fair use even when a defendant copies the entire work.[108] Furthermore, the third factor may not weigh against the defendant, so long as the defendant takes no more than what is necessary for use in the defendant's work.[109] The third-factor analysis uses as a qualitative element the importance of the material copied from the original work.[110] In *Harper & Row*, the Supreme Court agreed with the district court's evaluation of the copied work's qualitative character from the original.[111] The Supreme Court found that the copied work used "what was essentially the heart of the book," even though the portion in the copied work was a quantitatively insubstantial portion of the original work.[112]

Factor Four: Market and Value Effects

The fourth factor in a fair use analysis is the effect of the use upon the potential market for the copyrighted work or the effect on the value of the copyrighted work.[113] In analyzing the fourth factor, a court must look at the extent of the market harm that the copied work produced and whether the copier's

---

[103] 17 U.S.C. §107(3).
[104] Campbell, 510 U.S. at 587.
[105] *See Id.*; Seltzer v. Green Day, Inc., 725 F.3d 1170, 1178 (9th Cir. 2013).
[106] Campbell, 510 U.S. at 587-88.
[107] *Id.* at 587-88.
[108] Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510, 1526 (9th Cir. 1992).
[109] Seltzer, 725 F.3d at 1178.
[110] Harper & Row, Publishers, Inc v. Nation Enterprises, 471 U.S. 539, 564-65 (1985).
[111] *Id.*
[112] *Id.*
[113] 17 U.S.C. §107(4).

13

conduct in copying the original would impact the potential market for the original work.[114] The analysis must also take into account the market harm for derivative works.[115] According to the Supreme Court, the market for potential derivative uses of the original "includes only those that creators of original works in general develop or license others to develop."[116] Additionally, transformative uses weigh into the fourth-factor analysis because when a work is transformative, market substitution for the original work is less certain, and market harm may be less apparent.[117] According to the Second Circuit, courts should balance the fourth factor between the public's benefit if the use is allowed and the "personal gain the copyright owner will receive if the use denied."[118] Furthermore, the Second Circuit succinctly noted that "were a court automatically to conclude in every case that potential licensing revenues were impermissibly impaired simply because the secondary user did not pay a fee for the right to engage in the use, the fourth fair use factor would always favor the copyright holder."[119]

Standard of Review for the Fair Use Doctrine

"Fair use is a mixed question of law and fact," which an appellate court may decide if the lower court has found facts sufficient to evaluate the fair use factors.[120] Mixed questions differ from one another.[121] The Supreme Court has elaborated that when an issue happens to be somewhere between a legal standard and a fact, then the standard of review "often reflects which 'judicial actor is better positioned' to make the decision."[122] When it is necessary to expand on a legal standard in a mixed question, appellate courts should conduct a *de novo* review[123] of the lower court's decision.[124] However,

---

[114] Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 590 (1994).

[115] Harper & Row, 471 U.S. at 568.

[116] Campbell, 510 U.S. at 592.

[117] *Id.* at 591.

[118] Bill Graham Archives v. Dorling Kindersley Ltd., 448 F.3d 605, 613 (2d Cir. 2006).

[119] *Id.* at 614.

[120] Harper & Row, 471 U.S. at 560.

[121] U.S. Bank National Ass'n (ex rel. CWCapital Asset Management LLC) v. Village at Lakeridge, LLC, 138 S. Ct. 960, 967 (2018).

[122] *Id.*

[123] De novo review is when an "appellate court uses the trial court's record but reviews the evidence and law without deference to the trial court's rulings." *Appeal*, BLACK'S LAW DICTIONARY (11th ed. 2019).

[124] U.S. Bank National Ass'n, 138 S. Ct. at 967.

14

an appellate court should review the lower court's decision with deference[125] when the mixed question pertains to case-specific factual issues that require the lower court to weigh the evidence and "make credibility judgments."[126] Furthermore, the Supreme Court has stated that when a "district court has found facts sufficient to evaluate each of the statutory factors," then it is not necessary for an appellate court to remand for further factfinding, and the appellate court may conclude as a matter of law whether the copied work qualifies as fair use of the copyrighted work.[127] Additionally, the Ninth Circuit has concluded that when a record is sufficient to evaluate each fair use factor, then the appellate court "may reweigh on appeal the inferences to be drawn from that record."[128]

Oracle America, Inc. v. Google LLC

Background and Facts

After developing the Java platform, Sun Microsystems, Inc. ("Sun") released it in 1996.[129] Oracle America, Inc. ("Oracle") purchased Sun in 2010.[130] Before Java, computer programmers had to write different versions of the programs they created to run on different operating systems, such as Windows and macOS.[131] Java alleviated computer programmers' burden by allowing them to create a program that runs on all the different types of computers, through the Java platform, without rewriting the code for each operating system.[132] Java's motto is "write once, run anywhere."[133]

---

[125] Deferential review is "[a]n appellate standard granting relief from a lower court's. . . judgment only when earlier proceeding entailed an unreasonable application of clearly established law or a clearly unreasonable determination of the facts." *Review*, BLACK'S LAW DICTIONARY (11th ed. 2019).
[126] U.S. Bank National Ass'n, 138 S. Ct. at 967.
[127] Harper & Row, 471 U.S. at 560.
[128] Mattel, Inc. v. Walking Mountain Productions, 353 F.3d 792, 800 (9th Cir. 2003).
[129] Oracle America, Inc. v. Google Inc., 750 F.3d 1339, 1348 (Fed. Cir. 2014).
[130] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1186 (Fed. Cir. 2018).
[131] *See* Oracle America, Inc. v. Google Inc., 750 F.3d at 1348.
[132] *Id.*
[133] *Id.*

15

Sun created the Java language to write programs on the Java platform.[134] The Java language, like other programming languages, is a written language that is "human-readable."[135] However, the human-readable code, called "source code," is not readable by computer hardware.[136] Computers instead read what is called "object code," which is most commonly written in a binary language consisting of only 1s and 0s.[137] Thus, for a computer to read a computer program's source code, it must be converted into object code.[138]

Java 2 Standard Edition, commonly referred to as "Java SE," is the name of the Java platform at issue, which includes, among other things, the Java Application Programming Interface ("API").[139] "The Java API is a collection of 'pre-written Java source code programs for common and more advanced computer functions.'"[140] The APIs allow computer programmers to use the pre-written code to perform common functions without writing the code from scratch; they are shortcuts that computer programmers can use when writing a program.[141] The pre-written codes are organized into groups called "packages."[142] Each package contains "methods," which are the pre-written code for a specific function.[143] The methods within a package are sorted into "classes," the classes group similar methods together.[144] Likewise, the packages group related classes together.[145] A good analogy of the organization is: "Oracle's collection of API packages is like a library, each package is like a bookshelf in the library, each class is like a book on

---

[134] Oracle America, Inc. v. Google Inc., 872 F. Supp. 2d 974, 977 (N.D. Cal. 2012).
[135] *Id.*
[136] *Id.*
[137] *Id.*
[138] *Id.*
[139] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1186 (Fed. Cir. 2018).
[140] *Id.*
[141] *Id.*
[142] Oracle America, Inc v. Google Inc., 750 F.3d 1339, 1348-49 (Fed. Cir. 2014).
[143] *Id.* at 1349.
[144] Brief for Respondent at 4, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2020 WL 832871.
[145] *Id.*

16

the shelf, and each method is like a how-to chapter in a book."[146] The Java platform contains more than 6,000 methods that are sorted into over 600 classes that are grouped into 166 API packages.[147]

Packages contain two types of source code: the declaring code and the implementing code.[148] Computer programmers use a "call" to conjure the associated method.[149] The call is connected to the method by the declaring code.[150] If a call is not written exactly as required, it will not pull up the corresponding declaration, and thus, not work.[151] The declaring code (the "declaration" or the "header") is "the expression that identifies the prewritten function," and it "introduces the method body and specifics very precisely the inputs, name and other functionality."[152] The declaring code directs the computer to execute the implementing code associated with the specific declaring code.[153] The implementing code is the instructions to the computer to execute a specific function.[154] Implementing code can be written in various ways, unlike the declaring code.[155]

Oracle owns copyrights in Java SE, including in the API packages.[156] However, the Java language itself is free to use without permission.[157] For a programmer to write using the Java language, the programmer must use "62 classes (and some of their methods), spread across three packages within the Java library."[158] Without the use of the sixty-two classes, the Java language will not work.[159] Besides using the sixty-two classes, programmers can write their own prewritten code packages to perform various functions.[160] Oracle offers three different licenses on its copyrighted material: a General Public

---

[146] Oracle America, Inc. v. Google Inc., 750 F.3d at 1349.
[147] *Id.*
[148] *Id.*
[149] Brief for the Petitioner at 3, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2020 WL 104836.
[150] *Id.* at 5.
[151] *Id.*
[152] Oracle America, Inc. v. Google Inc., 750 F.3d at 1349.
[153] *Id.*
[154] *Id.*
[155] Brief for the Petitioner at 5, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956).
[156] Oracle America, Inc. v. Google Inc., 750 F.3d at 1350.
[157] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1187 (Fed. Cir. 2018).
[158] *Id.*
[159] *Id.*
[160] *Id.*

17

License, a Specification License, and a Commercial License.[161] The General Public License is an open-source license that allows programmers to use the declaring and implementing codes for free.[162] However, the programmer must give back to the public any innovations the programmer makes to the code.[163] The Specification License allows the licensee to use the declaring code and the structure of the API packages, but the licensee must write the implementing code.[164] Finally, the Commercial License allows businesses, in exchange for royalties, to use all of the copyrighted material while keeping any code the business creates secret.[165] Both the Specification and Commercial Licenses require that any programs created with the license be compatible with the Java platform.[166]

In 2005, Google and Sun began negotiations to license the use of the Java platform to create Google's mobile platform, Android.[167] The negotiations failed "because Google wanted device manufacturers to be able to use Oracle's APIs in Android for free with no limits on modifying the code, which would jeopardize the 'write once, run anywhere' philosophy."[168] Google nevertheless decided to use Java when creating its Android platform.[169] In doing so, Google copied the declaring code of thirty-seven API packages, which equates to 11,500 lines of copyrighted code, and the structure, sequence, and organization of the API packages.[170] When comparing Java's thirty-seven API packages, including the declaring and implementing code, to Google's thirty-seven API packages, the code that Google copied constitutes only three percent of the total code in Java's thirty-seven API packages.[171] Google's Android platform contains 168 API packages, which includes the thirty-seven copied packages.[172] Google used the specific thirty-seven API packages because it assumed that "programmers would want to find the same 37

---

[161] Oracle America, Inc. v. Google Inc., 750 F.3d at 1350.
[162] *Id.*
[163] *Id.*
[164] *Id.*
[165] *Id.*
[166] *Id.*
[167] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1187 (Fed. Cir. 2018).
[168] *Id.*
[169] *Id.*
[170] *Id.*
[171] Oracle America, Inc. v. Google Inc., 872 F. Supp. 2d 974, 979 (N.D. Cal. 2012).
[172] Oracle America, Inc. v. Google Inc., 750 F.3d at 1350.

18

sets of functionalities in the new Android system callable by the same names as used in Java."[173] Google's use of the thirty-seven sets allowed Android to be, in a way, interoperable with Java, meaning that programmers could use their knowledge of the Java calls to write programs both for desktops, which the Java platform runs on, and smartphones, which the Android platform runs on.[174] Even though Google used the Java language to write Android, Android is not compatible with the Java platform, which implies that applications and programs that a programmer writes for Android will not run on Java and vice versa.[175]

By using the thirty-seven API packages, Google "reimplemented" Java's interfaces.[176] Reimplementation is "[t]he process of writing new software to perform certain functions of a legacy product."[177] Free reimplementation of software interfaces is a long-standing practice that is important for "innovation and competition in software."[178] When a programmer reimplements an interface, it must maintain the structure, sequence, and organization of the interface.[179] Otherwise, the interface would be incompatible and not interoperable.[180]

In 2008, phones with the Android platform began selling to consumers.[181] Since their release, Android smartphones have grown in popularity and comprise "a large share of the United States market."[182] The Android platform is free for manufactures.[183] Instead of charging manufacturers to use the Android platform, Google generates revenue through advertising.[184] The release of the free Android

---

[173] *Id.*

[174] *See* Brief Amici Curiae of 83 Computer Scientists in Support of Petitioner at 3, Google LLC v. Oracle American Inc. (argued Oct. 7, 2020) (No. 18-956), 2020 WL 242506.

[175] Oracle America, Inc. v. Google Inc., 750 F.3d at 1351.

[176] Brief for the Petitioner at 7, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2020 WL 104836.

[177] *Id.*

[178] Brief Amici Curiae of 83 Computer Scientists in Support of Petitioner at 4, Google LLC v. Oracle American Inc. (argued Oct. 7, 2020) (No. 18-956).

[179] *Id.* at 12.

[180] *Id.*

[181] Oracle America, Inc. v. Google Inc., 750 F.3d 1339, 1351 (Fed. Cir. 2014).

[182] *Id.*

[183] *Id.*

[184] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1187 (Fed. Cir. 2018)

19

platform affected Oracle's licensing.[185] Amazon had initially licensed Java for its Kindle tablet, but then switched to Android when it released another generation of its Kindle tablet, and thereafter, Amazon leveraged Android against Oracle to get a large discount to use the Java platform again.[186]

Litigation Between Oracle America and Google in the Federal District and Circuit Courts

In 2010, a few months after Oracle acquired Sun, Oracle filed a suit against Google in the United States District Court for the Northern District of California, asserting seven claims of patent infringement and one claim of copyright infringement.[187] Oracle withdrew five of the seven patent infringement claims, and the remaining two claims went to a jury who ultimately found against Oracle.[188] Oracle's copyright claim asserted that Google copied Java API declarations, along with the structure, sequence, and organization of the API.[189]

The district court conducted a two-week trial.[190] The trial occurred in two phases.[191] The first phase was on copyrightability, copyright infringement, and equitable defenses.[192] The second phase was on patent infringement.[193] For phase one, the parties agreed that the Judge would decide the issues of copyrightability and equitable defenses asserted by Google, and the jury would decide the issues of copyright infringement and fair use.[194]

The jury received instructions to consider that the structure, sequence, and organization of the API packages at issue were copyrightable.[195] While that was not the actual legal conclusion, the

---

[185] *Id.* at 1188.
[186] *Id.*
[187] Petition for Writ of Certiorari at 7, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2019 WL 338902.
[188] *Id.* at 7-8
[189] *Id.* at 8.
[190] *Id.*
[191] Oracle America, Inc. v. Google, Inc., 872 F. Supp. 2d 974, 975 (N.D. Cal. 2012).
[192] *Id.*
[193] *Id.*
[194] *Id.*
[195] *Id.*

20

instruction intended to prevent an expensive retrial if the case underwent an appeal and reversal.[196]

Instead, the appellate court could simply reinstate the jury verdict.[197] The jury returned a verdict in favor

of Oracle, finding that "Google infringed Oracle's copyrights in the 37 Java packages," but the jury was

"deadlocked on Google's fair use defense."[198] Oracle then filed a motion for judgment as a matter of law

on the issue of fair use, which the district court denied.[199] Afterward, the district court issued a

preliminary decision, finding that the copied material in the thirty-seven API packages, which included

the declaring code and the structure, sequence, and organization, were not copyrightable because they

were a method of operation, and as such, are not copyrightable elements under 17 U.S.C. §102(b).[200] The

district court also found that there was only one way to write the declaring code, and therefore, the merger

doctrine applied, which barred copyright ownership to the expression.[201] Furthermore, the district court

"found that the declaring code was not protectable because 'names and short phrases cannot be

copyrighted.'"[202] Thereafter, the district court entered final judgment in favor of Google on the copyright

infringement claims relating to the thirty-seven API packages.[203] Oracle then appealed the rulings that the

court entered against it in the final judgment, and Google cross-appealed the rulings the court entered

against it in the final judgment.[204]

The appeal went to the U.S. Circuit Court of Appeals for the Federal Circuit because the suit

included patent claims and the Federal Circuit has jurisdiction over any suit that includes patent claims.[205]

The Federal Circuit, however, applied Ninth Circuit law to the appeal because, "'[w]hen the questions on

---

[196] *Id.*
[197] *Id.* at 975-76.
[198] Oracle America, Inc. v. Google Inc., 750 F.3d 1339, 1347 (Fed. Cir. 2014).
[199] *Id.*
[200] *Id.* at 1348, 1352; Petition for Writ of Certiorari at 8, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2019 WL 338902.
[201] Oracle America, Inc. v. Google Inc., 750 F.3d at 1352.
[202] *Id.*
[203] *Id.* at 1348.
[204] *Id.* For simplicity and because the copyrightability of thirty-seven API packages and fair use of those packages are the only issues granted certiorari in front of the U.S. Supreme Court, the other appealed issues will not be discussed.
[205] *Id.* at 1353; 28 U.S.C. §1295(a)(1).

21

appeal involve law and precedent on subjects not exclusively assigned to the Federal Circuit, the court applies the law which would be applied by the regional circuit.'"[206]

The Federal Circuit first looked at whether the declaring code was entitled to copyright protection under the merger doctrine and short phrases doctrine.[207] The appellate court concluded that the merger doctrine does not apply because there were alternative expressions available to Oracle when Oracle created the code.[208] Further, the appellate court stated that the district court failed to ask the correct question regarding the doctrine of short phrases.[209] "The question is not whether a short phrase or a series of short phrases can be extracted from the work, but whether the manner in which they are used or strung together exhibits creativity."[210] The appellate court reasoned that the declaring codes are not specific words or short phrases but are instead, 7,000 lines of declaring code, and the declaring codes expressed creativity in the selection and arrangement of the declarations.[211] The appellate court then concluded that the district court erred in applying the short phrases doctrine to find that the declaring codes are not protectable under copyright law.[212]

The appellate court then reviewed the district court's finding that the structure, sequence, and organization of the Java API packages are not copyrightable because they are a method of operation, which cannot be copyrighted under 17 U.S.C. §102(b), even if the API packages are creative and original.[213] The appellate court stated that the district court's reasoning was not based on Ninth Circuit law but, instead, on First Circuit law that the Ninth Circuit had not adopted.[214] The appellate court then provided the correct Ninth Circuit law, which states that "the structure, sequence, and organization of a

---

[206] Oracle America, Inc. v. Google Inc., 750 F.3d at 1353 (citing Atari Games Corp. v. Nintendo of America, Inc., 897 F.2d 1572, 1575 (Fed. Cir. 1990)).
[207] *Id.* at 1359.
[208] *Id.* at 1360-61.
[209] *Id.* at 1362.
[210] *Id.* at 1363.
[211] *Id.*
[212] *Id.*
[213] *Id.* at 1364.
[214] *Id.* at 1365-66.

22

computer program is eligible for copyright protection where it qualifies as an expression of an idea, rather than the idea itself."[215] Applying Ninth Circuit law, the appellate court found that the structure, sequence, and organization was creative and original and that the declaring code could have been written in a variety of ways while still achieving the same functions.[216] Therefore, the court concluded that 17 U.S.C. §102(b) does not bar copyright protection on the code.[217]

Finally, Oracle argued on appeal that the appellate court should find Google's use of the copyrighted material not to be fair use as a matter of law.[218] The appellate court found that neither the lower court nor the jury had made findings of fact that would allow the appellate court to assess the question of whether Google's use of the API packages was fair.[219] Namely, the appellate court found that there were disputed facts of whether Google's use of the copyrighted work was transformative, whether Google's desire to make the Android platform interoperable weighs into the fair use factors, and whether the release of Android harmed Java's market.[220] The appellate court remanded the proceedings to the district court for further proceedings to determine Google's fair use defense.[221]

After the Federal Circuit's opinion, Google petitioned for certiorari to the U.S. Supreme Court on the copyrightability issue.[222] The Supreme Court ultimately denied certiorari.[223]

Upon remand to the district court, the court had a second jury trial on the issue of fair use.[224] The jury returned a verdict in favor of Google on its fair use defense.[225] Thereafter, Oracle filed a motion for judgment as a matter of law, which the district court denied and entered final judgment in favor of

---

[215] *Id.* at 1366.
[216] *Id.* at 1368.
[217] *Id.*
[218] *Id.* at 1372.
[219] *Id.* at 1373.
[220] *Id.* at 1376-77.
[221] *Id.* at 1377.
[222] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1185 (Fed. Cir. 2018).
[223] Google, Inc v. Oracle America, Inc., 576 U.S. 1071, 135 S. Ct. 2887 (2015).
[224] Oracle America, Inc. v. Google LLC, 886 F.3d at 1185.
[225] *Id.*

23

Google.[226] Oracle then filed a renewed motion for judgment as a matter of law and a motion for a new trial.[227] The district court denied both motions, and Oracle appealed the denial along with the final judgment and prior denial of its motion for judgment as a matter of law.[228] Google also cross-appealed to preserve its claim as to copyrightability that the prior Federal Circuit opinion decided.[229]

Again, the appeal went to the U.S. Circuit Court of Appeals for the Federal Circuit.[230] On appeal, Oracle argued that all four of the statutory fair use factors weighed against a finding of fair use, namely that: (1) Google's use was solely commercial, (2) the nature of the copyrighted work is highly creative, (3) Google copied more than what was necessary to write in the Java language, and (4) Android harmed Oracle since Oracle's customers stopped licensing Java and switched to the Android platform.[231] The appellate court first looked at the standard of review for the issue of fair use and concluded:

> whether the court applied the correct legal standard to the fair use inquiry is a question [the appellate court] review[s] de novo, whether the findings relating to any relevant historical facts were correct are questions which [the appellate court] review[s] with deference, and whether the use at issue is ultimately a fair one is something [the appellate court] also review[s] de novo.[232]

The appellate court proceeded to state that, besides the jury's implied findings of historical fact, all the jury's other findings relating to fair use are advisory.[233]

Thereafter, the appellate court applied the four fair use factors.[234] The appellate court found that the first-factor analysis of the purpose and character of Google's use of the API packages was "overwhelmingly commercial" and that Google's use was not transformative, concluding that both of

---

[226] *Id.* at 1185-86.
[227] *Id.* at 1186.
[228] *Id.*
[229] *Id.*
[230] *See Id.* at 1185.
[231] *Id.* at 1191.
[232] *Id.* at 1191, 1193.
[233] *Id.* at 1196.
[234] *Id.*

24

those factors weighed against a finding of fair use.[235] The appellate court also reviewed the jury's implied finding that Google's use was not in bad faith for the first-factor analysis, and found that the factor does not weigh in Google's favor, only a finding of bad faith would weigh against Google.[236] As to factor two, the nature of the copyrighted work, the appellate court found that "reasonable jurors could have concluded that functional considerations [of the declarations and structure, sequence, and organization of the API packages] were both substantial and important," and concluded that factor two weighed in favor of a finding of fair use.[237] The factor three analysis on the amount and substantiality of the portion used, the appellate court found that even if the jury found that Google copied only a small fraction of Java SE, a reasonable jury could not find that "what was copied was qualitatively insignificant," and the court concluded that the third factor is at best neutral to a finding of fair use, but that it could also weigh against a finding of fair use.[238] Finally, in the fourth-factor analysis, the appellate court found evidence of actual market harm, which no reasonable jury could have concluded otherwise, and that there was potential market harm for derivative works.[239] The appellate court, therefore, concluded that the fourth factor weighed against a finding of fair use.[240] Thereafter, the appellate court weighed all four factors and concluded that Google's use of the copyrighted material was not fair as a matter of law and remanded the proceeding for a trial to determine the amount of damages.[241]

After the Federal Circuit found against Google's fair use defense, Google petitioned the Supreme Court for a writ of certiorari.[242] The Supreme Court granted certiorari on November 15, 2019, and the Supreme Court heard oral arguments on October 7, 2020.[243]

---

[235] *Id.* at 1198, 1202, 1204.
[236] *Id.* at 1203-04.
[237] *Id.* at 1205.
[238] *Id.* at 1207.
[239] *Id.* at 1209-10.
[240] *Id.* at 1210.
[241] *Id.* at 1211.
[242] *See* Petition for Writ of Certiorari, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), 2019 WL 338902.
[243] Google LLC v. Oracle America, Inc., 140 S. Ct. 520 (2019); Transcript of Oral Argument at 1, Google LLC v. Oracle America, Inc. (argued Oct. 7, 2020) (No. 18-956), https://www.supremecourt.gov/oral_arguments/argument_transcripts/2020/18-956_kifl.pdf.

25

III.   Discussion

What the Federal Circuit Likely Got Right

Although the U.S. Supreme Court will likely find that the Federal Circuit misinterpreted some case law and misconstrued some of the facts to reach legal conclusions, the U.S. Supreme Court will likely find that the Federal Circuit was correct on some of its conclusions. The Federal Circuit's holding that Java's declaring code and the structure, sequence, and organization of the APIs were copyrightable was likely the correct conclusion, at least as to some aspects they are copyrightable. The Federal Circuit appears to give the structure, sequence, and organization of the APIs full copyright protection, but it should only be entitled to a thin copyright. The structure, sequence, and organization of the APIs are similar to the selection and arrangement of the telephone directory in *Feist Publications, Inc v. Rural Telephone Service Co.*[244] In both the telephone directory and the structure, sequence, and organization of the APIs, the creator decided how to organize the information.[245] However, in the case of the telephone directory, the organizational scheme of the directory was in no way original and was commonplace, whereas the organizational scheme of the Java APIs was original, and others creating a similar organizational scheme would likely structure it differently. The Federal Circuit likely overemphasized the creativeness in designing the structure, sequence, and organization of the APIs. On the other hand, the declaring code is far more creative than the APIs organizational scheme, and thus, likely deserves more than a thin copyright as to any of the declaring code's non-functional elements. As stated by the Federal Circuit, Oracle had an unlimited amount of options when creating its declaring code; for example, instead of using the word "max" in its code, it could have used the word "larger" or "greater."[246]

The U.S. Supreme Court will likely find that the Federal Circuit was also correct to apply the Second Circuit's "abstraction-filtration-comparison" test to the copyrightability of the declaring code and

---

[244] 499 U.S. 340, 349 (1991).
[245] *Id.* at 363; Oracle America, Inc. v. Google Inc., 750 F.3d 1339, 1363 (Fed. Cir. 2014).
[246] Oracle America, Inc. v. Google Inc., 750 F.3d 1339, 1361 (Fed. Cir. 2014).

26

the structure, sequence, and organization of the APIs. The Federal Circuit stated that the District Court failed to apply the abstraction-filtration-comparison test to the copyrightability question, but the Federal Circuit itself failed to go through the analysis.[247] The U.S. Court of Appeals for the Ninth Circuit endorsed the abstraction-filtration-comparison test as the proper approach, and because the Federal Circuit is required to apply Ninth Circuit law in the case at hand, the Federal Circuit should have performed the test's lengthy analysis.[248] Because the Federal Circuit failed to properly analyze the copied material at issue under the abstraction-filtration-comparison test, the Federal Circuit may have allowed aspects of the declaring code and structure, sequence, and organization of the APIs to be deemed protected under copyright when they should have been deemed uncopyrightable. Computer programs are in essence, functional, but that does not mean that creativity does not go into designing a computer program. The Third Circuit's view that functional aspects of a computer program are copyrightable, so long as there are multiple ways to express it, is overly broad and would allow almost all functional aspects to be copyrightable, which would inhibit creativity. The First and Sixth Circuit's view that functional aspects are not protectable by copyright is too narrow. Many functional aspects in a computer program have underlying creative non-functional aspects deemed uncopyrightable if using this approach. The Ninth Circuit's abstraction-filtration-comparison is in between the two opposing views. Since the abstraction-filtration-comparison test allows certain creative non-functional aspects within functional aspects of a computer program to be copyrightable, it appears to be a better approach to the copyrightability of elements in a computer program.

The U.S. Supreme Court will likely find that the Federal Circuit's conclusion was accurate when it found that the district court performed the merger analysis on the incorrect timeframe. The merger analysis does not apply at the time an infringer copies the work, but at the time the author created the work.[249] In the Federal Circuit's conclusion, it was correct in stating that the merger analysis did not apply

---

[247] *Id.* at 1358.
[248] *See* Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).
[249] Oracle America, Inc. v. Google Inc., 750 F.3d at 1361.

27

because there were multiple ways that the declaring code could have been expressed at the time it was created. The district court concluded that there was only one way to write the declaring code, but the district court failed to consider that the programmer could have expressed it in multiple ways when the programmer created the code. The district court's conclusion is akin to saying that there is only one way of pressing the keys to type the word "computer" on a QWERTY keyboard; it is the only way because the creator set up the keys in a particular way when the creator created the QWERTY keyboard. However, suppose a person were to look at how to press the keys to create the word "computer" at the time of creation. In that case, the creator could have set up the keys in a variety of combinations. Only after the creator expressed the QWERTY keyboard's keys in the creator's particular way did the way of pressing the keys to type the word "computer" become the only way to do it.

What the Federal Circuit Likely Got Wrong

While the U.S. Supreme Court will likely find that the Federal Circuit may have gotten some things correct, the Supreme Court will likely find that the Federal Circuit also got some things wrong. First, the Supreme Court will likely find that the Federal Circuit's analysis of the law on the standard of review for fair use and the court's conclusion on that law was starkly incorrect. The Federal Court appears to have incorrectly analyzed a case where the Supreme Court explained how to determine a mixed question of law and fact as it applied *in a bankruptcy proceeding*.[250] The Federal Court incorrectly stated that the Supreme Court opinion in *U.S. Bank National Ass'n ex rel. CWCaptial Asset Management LLC v. Village at Lakeridge, LLC* is "the standard of review . . . in connection with any mixed question of law and fact."[251] However, a person reading the Supreme Court's opinion could identify that certain aspects of the opinion related only to bankruptcy proceedings.[252] For example, when the Court discussed "historical"

---

[250] Oracle America, Inc. v. Google LLC, 886 F.3d 1179, 1192 (Fed. Cir. 2018).
[251] *Id.*
[252] *See* U.S. Bank National Ass'n *ex rel.* CWCapital Asset Management LLC v. Village at Lakeridge, LLC, 138 S. Ct. 960 (2018).

28

facts, it identified that it was regarding bankruptcy.[253] Because the Federal Circuit misconstrued the Supreme Court's opinion, it deemed that only the historical facts were entitled to a deferential review while any other facts that the jury found were merely advisory. This is incorrect as all factual findings by a jury on review should be treated with deference. The Federal Circuit's standard of review allowed it to make its own findings of fact, which ultimately led to a finding that Google's use of the copyrighted material was not fair.

The U.S. Supreme Court will likely find that the Federal Circuit's fair use analysis was mistaken. In the Federal Circuit's analysis of the first factor, the purpose and character of the use, the court looked at commerciality and transformative use. While the Federal Circuit's finding that Google's use was commercial was likely correct, it placed too much emphasis on its commerciality, stating that it was "overwhelmingly commercial."[254] The downfall of not merely the first-factor analysis but the entire fair use analysis was the Federal Circuit's finding that Google's use was not transformative. The Federal Circuit started by stating that, according to Ninth Circuit law, whether a work is transformative is a question of law[255]—that is incorrect. The Ninth Circuit never analyzed transformation as a question of law. The Ninth Circuit merely stated that whether a work is a parody is a question of law.[256] In deciding that Google's use was not transformative as a matter of law, the Federal Circuit gave flawed reasoning. One reason the Federal Circuit provided for why Google's use was not transformative was that "it does not fit within the uses listed in the preamble to [17 U.S.C.] §107,"[257] however, that list is not all-inclusive.[258] Another reason the Federal Circuit determined Google's use was not transformative was that it served the same purpose in both platforms.[259] This reason is unpersuasive. The Federal Circuit failed to

---

[253] "[A] bankruptcy court evaluating insider status must make findings of what we have called "basic" or "historical" fact—addressing questions who did what, when or where, how or why." *Id.* at 966.

[254] Oracle America, Inc. v. Google LLC, 886 F.3d at 1198.

[255] *Id.* at 1199.

[256] "The issue of whether a work is a parody is a question of law, not a matter of public majority opinion." Mattel Inc. v. Walking Mountain Productions, 353 F.3d 792, 801 (9th Cir. 2003).

[257] Oracle America, Inc. v. Google LLC, 886 F.3d at 1199.

[258] "[T]he fair use of a copyrighted work, . . . for purposes *such as* criticism, comment, news reporting, teaching . . . , scholarship, or research, is not an infringement of copyright." 17 U.S.C. §107 (emphasis added).

[259] Oracle America, Inc. v. Google LLC, 886 F.3d at 1199.

29

consider the work as a whole, and instead, looked only at the copied material. Google reimplemented the declaring code and structure, sequence, and organization of API packages from the Java platform, which platform Sun created to work on desktop computers.[260] Google used what it copied in a transformative way, including using its own implementing code, which allowed the copied material to work in an environment that they could not work before, the touchscreen mobile smartphone.[261] Furthermore, the Federal Circuit failed to consider Google's stated purpose for using the declaring code and the structure, sequence, and organization of thirty-seven of Oracle's API packages, to attain a degree of interoperability with the Java platform, which would make it easier for programmers to use both the Java and Android platforms. Reasonable jurors could have found that Google's use was transformative and that the transformative use was enough to overcome that the use was commercial. The first-factor analysis should have weighed in favor of a finding of fair use.

While the U.S. Supreme Court will likely find that the Federal Circuit's analysis on the second factor, the nature of the copyrighted work, is correct, the Supreme Court will likely find that the Federal Circuit's comment that the factor "has less significance to the overall analysis" is not.[262] The Supreme Court has stated that courts are not to treat the statutory factors in isolation and that courts should consider all factors and "the results weighed together, in light of the purposes of copyright."[263] The second factor may, at times, have less weight than the other factors, but it is not less significant than the other factors. If the factor was not important, why would Congress include it as a factor that courts must consider when determining fair use?

The U.S. Supreme Court will likely find that the Federal Circuit's analysis of the third factor, the amount and substantiality of the copyrighted work used, is without merit. Instead of looking at the portion used from the copyrighted work as a whole, the Federal Circuit looked at the amount of code necessary to

---

[260] Oracle America, Inc. v. Google Inc., No. C 10-03451 WHA, 2016 WL 3181206, at *9 (N.D. Cal. June 8, 2016), *rev'd and remanded sub nom.* Oracle Am., Inc. v. Google LLC, 886 F.3d 1179 (Fed. Cir. 2018).
[261] *Id.*
[262] Oracle America, Inc. v. Google LLC, 886 F.3d at 1205.
[263] Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569, 578 (1994).

30

write in the Java language, finding that Google copied more than necessary.[264] However, that is not what a court looks at for this factor. Google copied only a small amount of code when looked at in proportion to the total amount of code in the Java platform.[265] Although the Federal Circuit found this factor neutral, the court also said that it could also weigh against a finding of fair use. Because the amount of code that Google copied was small in proportion to the total amount of code in the Java platform, and, according to Google, it copied no more than what was necessary "to preserve inter-system consistency in usage," even if the factor does not weigh in favor of a finding of fair use, the factor should not weigh against a finding of fair use; thus, the factor should be neutral in the analysis.[266]

Finally, the U.S. Supreme Court will likely find that the Federal Circuit's analysis of the fourth factor, the effect on the market for the copyrighted work, is overstated. The Federal Circuit stated that "no reasonable jury could have concluded that there was no market harm to Oracle from Google's copying."[267] While there was evidence that Android produced actual market harm to the Java platform, as shown by Amazon renegotiating a better deal on a license with Oracle, a reasonable jury could have found that the market harm was not severe. Furthermore, the Federal Circuit reasoned that there was potential market harm as Java has the right to enter the smartphone market.[268] However, that market was a market that they already attempted to enter. A different version of the Java platform, Java Micro Edition,[269] was in mobile phones, but revenue from that platform had declined.[270] Furthermore, because Sun made the Java API free under an open-source license, the jury could have found that there could be

---

[264] Oracle America, Inc. v. Google LLC, 886 F.3d at 1206.

[265] *Id.*

[266] *Id.*

[267] *Id.* at 1209.

[268] *Id.* at 1210.

[269] Java Micro Edition ("Java ME") is a "derivative version of the Java platform for use on mobile devices." Oracle America, Inc. v. Google, Inc., 750 F.3d 1339, 1350 (Fed. Cir. 2014).

[270] Oracle America, Inc. v. Google Inc., No. C 10-03451 WHA, 2016 WL 3181206, at *10 (N.D. Cal. June 8, 2016), *rev'd and remanded sub nom.* Oracle Am., Inc. v. Google LLC, 886 F.3d 1179 (Fed. Cir. 2018).

31

no market harm as the Java API was already free.[271] Therefore, the Federal Circuit should have found only minimal market harm to Oracle.

Reweighing all factors, the U.S. Supreme Court will likely find that the Federal Circuit should have found that Google's use of the declaring code and the structure, sequence, and organization of thirty-seven of Oracle's API packages as fair.

IV.   Conclusion

Free reimplementation of software interfaces has been an important long-standing practice for software innovation. Given this long-standing practice in the software industry, the Supreme Court likely will not completely upend the industry by ruling that the declaring code and the structure, sequence, and organization of API packages are protected under copyright and that Google's use of those was not fair.

The Supreme Court could simply hold that the Federal Circuit did not properly apply the correct standard of review for the fair use defense and remand the case to the appellate court to apply the correct standard of review. However, this ruling is not likely as it would allow other similar disputes to percolate up to the Supreme Court and provide no more clarity to the uncertainty faced in the field of copyrights in computer programming.

Given the line of questioning asked by the Justices during the oral argument, the Supreme Court will likely find that the declaring code and the structure, sequence, and organization of thirty-seven of Oracle's API packages are protectable under copyright law. However, the Supreme Court will likely find that Google's use of the copyrighted material was fair. This ruling would not wholly upend the software industry as courts could still find fair use when programmers reimplement software interfaces. However, it would create more litigation to determine whether a programmer's use of the reimplemented interfaces was fair.

---

[271] *Id.*

32