

# Copyright in Code: Supreme Court Hears Landmark Software Case in *Google v. Oracle*

October 21, 2020

In what observers have [hailed](#) as the “copyright case of the century,” an eight-member Supreme Court heard arguments on October 7, 2020, in *Google LLC v. Oracle America Inc.*, a long-running intellectual-property dispute between the two tech giants. Along with the billions of dollars [at stake](#) between the parties, the Court’s decision in *Google v. Oracle* could have far-reaching implications for software companies, the broader technology industry, and other copyright-intensive industries. Reflecting these stakes, the Supreme Court received over [70 amicus briefs](#) from industry, advocacy groups, academics, and other stakeholders, ranging from [computer scientists](#) and [small software startup](#) firms to [IBM](#), [Microsoft](#), and the [Motion Picture Association](#). This Sidebar reviews the legal doctrines at issue in *Google*, the facts of the dispute, the parties’ arguments, and the potential implications of the Court’s decision for Congress.

## Software Copyright Basics

Copyright law grants certain [exclusive legal rights](#) to authors of original [creative works](#), such as books, music, fine art, and architecture. At least [since 1980](#), U.S. copyright law has protected computer programs as a type of [literary work](#). Applying legal principles originally crafted for books to computer code has not always been a [straightforward task](#), in part because computer programs are more functional than other copyrightable subject matter. Courts have [long wrestled](#) with the appropriate scope of copyright protection in computer code. When the Supreme Court last tried to weigh in on software copyright in the 1990s, it [divided 4-4](#) and therefore issued no precedential decision. Given that the Court heard arguments in *Google* with eight Justices presiding, there is at least a possibility of a 4-4 split in this case as well, although the probability of such an outcome remains unclear.

Three key copyright doctrines affect the scope of copyright protection for computer code. The first is the *idea/expression dichotomy*, codified in Section 102(b) of the Copyright Act, which [states](#) that copyright protection does not “extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery.” This doctrine derives from the Supreme Court’s 1880 decision in *Baker v. Selden*, which held that the copyright in a book describing a system of accounting extended only to the author’s particular description of that system (the book’s “expression”) and not to the accounting system itself (the book’s “idea”). The second doctrine, known as *merger*, is a corollary of the idea-expression distinction. When there are only a few ways to express an idea, the expression is said to “[merge](#)” with the

Congressional Research Service

<https://crsreports.congress.gov>

LSB10543

idea, and neither is copyrightable. One central purpose of both doctrines is to [prevent](#) the use of copyright to monopolize general ideas or functional systems.

The third doctrine is *fair use*, which [permits](#) limited uses of copyrighted works that would otherwise be infringements, such as using portions of a copyrighted work in a parody or book review. To determine whether a use is fair, courts consider [a number of factors](#), including (1) the purpose and character of the use, (2) the nature of the original work, (3) the amount and substantiality of what was copied, and (4) the commercial harm from the use on the potential market for the original work. As part of the first factor, courts also consider whether the alleged fair use is “[transformative](#),” that is, whether it adds new expression, has a different purpose, or alters the original work with new expression or meaning. Applications of fair use are wide-ranging; under “the common law tradition of fair use adjudication,” courts [rely](#) on fair use to avoid “rigid application” of copyright liability when it would “stifle the very creativity which [copyright] is designed to foster.”

## The Dispute in *Google v. Oracle*

The dispute in *Google v. Oracle* concerns the Android operating system for smartphones. In developing Android, Google [copied](#) certain elements of Oracle’s Java programming language and platform. In particular, Java [contains](#) thousands of *methods*, sometimes referred to collectively as the application programming interface (API). Methods [are](#) modules that application developers can invoke (or “call”) to perform certain functions, rather than writing basic code from scratch. Java [groups](#) related methods into classes, and related classes into packages. For example, Java’s “*Math*” class includes, among other related methods, the “*max*” [method](#), a pre-built function that Java programmers can use to output the greater of two input values. Thus, a programmer can [call](#) `java.lang.Math.max(x, y)` to determine whether x or y is a larger number (and output that number), rather than independently writing code to perform the function. In building Android, Google [copied](#) the “declaring code” of 37 of the Java API’s 166 packages. The declaring code [includes](#) the name for the function (in this example, “max”) and its syntax, as well as its place within Java’s taxonomy of methods (in this example, within the “math” class). Google independently [wrote](#) Android’s “implementing code,” the operative code that performs the method. In all, Google [copied](#) over 11,000 lines of code (of about [15 million](#) in Android) so that developers writing applications for Android could rely on the Java calls with which they were already familiar.

Oracle [sued](#) Google in 2010, claiming both patent and copyright infringement, and seeking billions in damages. The copyright claims were tried to a jury, which [found](#) that Google infringed but deadlocked on Google’s fair use defense. The district court judge, however, set aside the infringement verdict, [holding](#) that the declaring code at issue—including the Java API’s structure, sequence, and organization (SSO)—was not copyrightable under Section 102(b) and the merger doctrine. The Federal Circuit [reversed](#), holding that the declaring code and the API’s SSO were copyrightable, and that neither Section 102(b) nor merger applied. On remand, a second jury [found](#) that Google’s use of the declaring code was fair. Oracle again appealed, and the Federal Circuit [reversed](#), holding that Google’s use of Java’s declaring code and the API SSO was not fair as a matter of law.

The Supreme Court granted certiorari [to address](#) (1) “[w]hether copyright protection extends to a software interface” and (2) “[w]hether, as the jury found, [Google’s] use of a software interface in the context of creating a new computer program constitutes fair use.” On its own accord, the Supreme Court [ordered](#) supplemental briefing to address a third issue, the “appropriate standard of review” for a jury verdict on fair use, “including but not limited to the implications of the Seventh Amendment, if any, on that standard.” (The [Seventh Amendment](#) guarantees the right to a trial by jury in certain civil cases, including [copyright cases](#) seeking monetary damages.)

## The Parties' Arguments

Google first argues that the merger doctrine [controls](#) the case, and precludes copyright protection when there are only a few ways to perform a particular function. Google [characterizes](#) the declaring code as “an interface connecting the operating system to commands in applications written by software developers.” It [casts](#) its use as part of a long-settled practice of software “reimplementation,” where a new market entrant generally writes new code but reuses a “limited number of instructions” to recreate an interface already known to users. On this view, because a developer must write the declaring code in a certain way to respond to the specific calls already known to Java developers, the code is [not copyrightable](#) under the merger doctrine. Alternatively, Google [argues](#) that there is no basis to overturn the jury’s verdict on fair use. On the contrary, Google [urges](#) that a reasonable jury could have found that Android represents a transformative use of the declaring code, the copying represented a small fraction of the Java API code, or that the resulting market harm was limited because Java does not compete with Android in the smartphone market.

In response, Oracle [emphasizes](#) that many original creative choices went into the 11,330 lines of code that Google copied, including how Java’s creators named, structured, and organized the thousands of methods of the Java API. Oracle [argues](#) that Google’s Section 102(b) and merger arguments are thus meritless: because the declaring code could have been written in many ways, it is copyrightable, and Section 102(b) does not withdraw copyright protection just because the methods, like most computer code, are functional. As to merger, Oracle notes that Google could have written new methods from scratch to perform the same functions, but chose to copy in order to “[leverage](#)” Java’s popularity with developers without obtaining a license on Oracle’s terms. As to fair use, Oracle [argues](#) that the ultimate conclusion of fair use is a legal issue that courts may review de novo, and that Google’s competing, nontransformative commercial use is the antithesis of a fair one.

In its supplemental briefing, Google [urges](#) that the appropriate standard of judicial review of a fair use verdict is highly deferential: after construing all factual disputes in favor of the verdict, the court may overturn a general jury verdict only if *no* reasonable jury could have found the use was fair. Because fair use is a mixed question of law and fact, Google [argues](#) that it was appropriate to commit it to a jury, noting that both parties agreed to do so. For its part, Oracle [asserts](#) that the appropriate standard of review is de novo. Relying on *Harper & Row v. Nation Enterprises*, Oracle [argues](#) that although the court must defer to the jury’s fact-finding, the ultimate question of whether Google’s copying is fair is a legal judgment that courts determine de novo. That said, Oracle [maintains](#) that the standard of review “makes no difference” in this case because no reasonable jury could find that Google’s use was fair.

## Implications for Congress

Depending on how broadly it rules and the issues it chooses to address, the Court’s ruling in *Google v. Oracle* could have sweeping ramifications for the technology industry. For example, a ruling on the merger doctrine may restrict or broaden competitors’ ability to copy functional elements of computer software or interfaces, potentially imperiling existing business models or opening up new avenues for reuses of existing software. A ruling on just the fair use issue, however, may be narrower in its effects beyond the litigants, because fair use typically requires a “[case-by-case](#)” analysis. That said, how the Court approaches the fair use analysis and the factors it chooses to emphasize will likely [influence](#) the development of the doctrine and the analyses by lower courts in future cases, potentially affecting the scope of permissible uses of copyrighted works in contexts far removed from computer programs.

In some ways, *Google* can be viewed as part of a long history of courts and Congress seeking to adapt copyright law to new media and changing technological contexts. For example, in the early 20<sup>th</sup> century, the Supreme Court [grappled](#) with whether copyright protection for musical works reached player piano rolls, with Congress subsequently [overruling](#) the Court’s holding that such rolls were not “copies” of the

musical work via statute. Copyright protection for architectural works was not generally available until 1990. Just a few years ago, Congress extended federal copyright to early sound recordings via the Music Modernization Act.

Copyright law is a statutory creation; Congress thus has the power to change the scope of software copyright should it disagree with the decision in *Google v. Oracle*. For example, Congress could amend Section 102(b) to clarify how the idea/expression dichotomy applies to computer programs, resolving the tension between the general copyrightability of computer programs with Section 102(b)'s prohibition of copyright in "method[s] of operation." Congress could specify, for example, whether and when declaring code, software interfaces, graphical user interfaces, and nonliteral aspects of computer programs are copyrightable. Although Congress has broad authority over the scope of copyright, such legislation must comport with constitutional limitations—such as the Takings Clause of the Fifth Amendment—and the United States' international-treaty commitments, such as those contained in the Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS).

## Author Information

Kevin J. Hickey  
Legislative Attorney

---

## Disclaimer

This document was prepared by the Congressional Research Service (CRS). CRS serves as nonpartisan shared staff to congressional committees and Members of Congress. It operates solely at the behest of and under the direction of Congress. Information in a CRS Report should not be relied upon for purposes other than public understanding of information that has been provided by CRS to Members of Congress in connection with CRS's institutional role. CRS Reports, as a work of the United States Government, are not subject to copyright protection in the United States. Any CRS Report may be reproduced and distributed in its entirety without permission from CRS. However, as a CRS Report may include copyrighted images or material from a third party, you may need to obtain the permission of the copyright holder if you wish to copy or otherwise use copyrighted material.