No. 13-298

IN THE
Supreme Court of the United States

ALICE CORPORATION PTY. LTD.,
Petitioner,
v.
CLS BANK INTERNATIONAL
AND CLS SERVICES LTD.,
Respondents.

On Writ of Certiorari to the United States Court of Appeals for the Federal Circuit

BRIEF OF SOFTWARE FREEDOM LAW CENTER, FREE SOFTWARE FOUNDATION, AND OPEN SOURCE INITIATIVE AS *AMICI CURIAE* IN SUPPORT OF RESPONDENTS

EBEN MOGLEN
Counsel of record
MISHI CHOUDHARY
JONATHAN D. BEAN
Software Freedom Law Center
1995 Broadway, 17th Floor
New York, New York 10023
moglen@softwarefreedom.org
212-461-1900

February 27, 2014

QUESTION PRESENTED

Whether claims to computer-implemented inventions—including claims to systems and machines, processes, and items of manufacture—are directed to patent-eligible subject matter within the meaning of 35 U.S.C. § 101 as interpreted by this Court?

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases

Constitutions, Statutes, and Rules

Other Authorities

INTEREST OF AMICI CURIAE

Software Freedom Law Center

Much of the world's most important and most commercially significant software is distributed under copyright licensing terms that give recipients freedom to copy, modify and redistribute the software ("free software").[1] One could not send or receive e-mail, surf the World Wide Web, perform a Google search or take advantage of many of the other benefits offered by the Internet without free software. Indeed, this brief was written entirely with free software word processors, namely GNU Emacs and LaTeX, each of which are not just competitive with or superior to non-free software programs, but which also provide their users with the freedom to improve the program to fit their needs and reflect their desires.

The Software Freedom Law Center ("SFLC") is a not-for-profit legal services organization that provides legal representation and other law-related services to protect and advance free software. SFLC provides pro bono legal services to non-profit free software developers and also helps the general public better understand the legal aspects of free software. SFLC has an interest in this matter because the decision of this Court will have a significant effect on the rights of the

[1]Pursuant to Sup. Ct. R. 37.6, amici note that no counsel for a party authored this brief in whole or in part, and no counsel or party made a monetary contribution intended to fund the preparation or submission of this brief. No persons other than amici curiae and their counsel made a monetary contribution to its preparation or submission. Petitioners and Respondents have consented to the filing of this brief through blanket consent letters filed with the Clerk's Office.

free software developers and users SFLC represents. More specifically, SFLC has an interest in ensuring that limits are maintained on the reach of patent law so that free software development is not unreasonably and unnecessarily impeded.

Free Software Foundation

This brief is filed on behalf of the Free Software Foundation, a charitable corporation with its main offices in Boston, Massachusetts. The Foundation believes that people should be free to study, share and improve all the software they use and that this right is an essential freedom for users of computing. The Foundation has been working to achieve this goal since 1985 by directly developing and distributing, and by helping others to develop and distribute, software that is licensed on terms that permit all users to copy, modify and redistribute the works, so long as they give others the same freedoms to use, modify and redistribute in turn. The Foundation is the largest single contributor to the GNU operating system (used widely today in its GNU/Linux variant for computers from PCs to supercomputer clusters). The Foundation's GNU General Public License is the most widely used free software license, covering major components of the GNU operating system and hundreds of thousands of other computer programs used on hundreds of millions of computers around the world. The Foundation strongly rejects the use of patent law to control the making and distribution of software. It believes that the misapplication of patent law to computer software prevents the development, distribution and use of free/libre software, and therefore endangers users' control of their digital activities.

Open Source Initiative

The Open Source Initiative ("OSI") is a not-for-profit organization that supports and promotes the open source movement. Open source is a development method for software that harnesses the power of distributed peer review and development to build better software. The resulting software is globally available, and provides more user flexibility and reliability, at lower cost than traditional, centralized software development methods. Founded in 1998 by some of the people who coined the term "open source", OSI promotes open source development, advocates for the open source community, and maintains the Open Source Definition that helps determine whether a project is open source. OSI's membership is global, and includes individual developers, affiliated open source projects, and corporate sponsors who participate in and benefit from open source. OSI has an interest in this matter because the decision of this Court will have a significant effect on the rights and activities of the developers and users who make up the open source movement. In particular, OSI has an interest in limiting the reach of patent law so that open source software development is not unreasonably impeded.

SUMMARY OF ARGUMENT

As this Court has held consistently, any subject matter that risks pre-empting free use of laws of nature, algorithms, or abstract ideas is not eligible for patenting. This Court has repeatedly stated that, in determining the patentability of processes, the presence of a particular machine or apparatus or transformation

of matter is a strong clue that the process claimed is patent-eligible under §101. *Cochrane* v. *Deener*, 94 U.S. 780, 788 (1877); *Gottschalk* v. *Benson*, 409 U.S. 63, 70–71 (1972); *Parker* v. *Flook*, 437 U.S. 584, 588 n. 9 (1978); *Diamond* v. *Diehr*, 450 U.S. 175, 184 (1981); *Bilski* v. *Kappos*, 130 S. Ct. 3218, 3227 (2010); *Mayo Collaborative Services* v. *Prometheus Laboratories, Inc.*, 132 S. Ct. 1289, 1303 (2012).

The present case raises the question how to determine patent eligibility for computer-implemented inventions only, a narrower category of patent applications than those considered in *Bilski*. In this narrower category of cases, the Court should adopt the "machine or transformation" test as the bright line. In the more than sixty years since the adoption of the 1952 Patent Act amendments, the Court has never faced a patent application for a computer-implemented invention that failed the "machine or transformation" test and yet fell within the scope of §101. Speculation about such possible cases should not prevent the clarity that adoption of a bright-line test would bring.

The "built-in" accommodation between copyright law and the First Amendment, *see Eldred* v. *Ashcroft*, 537 U.S. 186, 219 (2003), is present, but in a different form, in patent law. In *Eldred*, the Court held that the idea/expression distinction and the fair use principle are constitutionally required to prevent collision between copyright and the First Amendment. The same required function is performed in patent law by the exemption of all subject matter that pre-empts free use of laws of nature, algorithms and abstract ideas. In cases involving computer software, the risk of creating statutory monopolies on ideas is particularly high, because computer programs, as the Court has held, are abstract ideas without physical embodiment. The

Court must construe the Patent Act to avoid constitutional infirmity, and it does so in this context by applying the "machine or transformation" test to such applications seeking statutory monopolies for computer-implemented inventions.

ARGUMENT

I. Processes Are Not Patentable If They Are Implemented Solely Through Computer Software, Without a Specialized Machine, or Transformation of Matter, As This Court Has Uniformly Held

As Justice Breyer noted in *Bilski*, 130 S. Ct., at 3258:

> [A]lthough the text of §101 is broad, it is not without limit. . . . In particular, the Court has long held that '[p]henomena of nature, though just discovered, mental processes and abstract intellectual concepts are not patentable' under §101, since allowing individuals to patent these fundamental principles would 'wholly pre-empt' the public's access to the basic tools of scientific and technological work.

(internal citations omitted) (quoting *Benson*, 409 U.S., at 67, 72) (citing *Diehr*, 450 U.S., at 185; *Diamond* v. *Chakrabarty*, 447 U.S. 303, 309 (1980)).

In keeping with this principle, the Court has recognized in an unbroken series of cases extending over

more than a century that patents should not be allowed to preempt the fundamental tools of discovery which should remain "free to all ... and reserved exclusively to none." *Funk Brothers Seed Company* v. *Kalo Inoculant Company*, 333 U.S. 127, 130 (1948). Patent eligibility at the constitutional limit cannot be made the handmaiden of a clever draftsman. The test articulated for patentability must take no account of the terms in which claims are posed. The Court has stated that "[t]ransformation and reduction of an article to a different state or thing is the clue to the patentability of a process claim that does not include particular machines." *Benson*, 409 U.S., at 71 (internal quotation marks omitted). Summarizing this history, *Flook* remarked that the Court had "only recognized a process as within the statutory definition when it either was tied to a particular apparatus or operated to change materials to a 'different state or thing.'" 437 U.S., 588 n. 9 (quoting *Cochrane* v. *Deener*, 94 U.S, at 787–88). In *Diamond* v. *Diehr*, 450 U.S., at 184, this Court once again applied the "machine or transformation" test regarding the patentability of processes under §101. Though the Court has repeatedly cautioned that the "machine or transformation" test is not the *sole* expression of the limits of §101, *Bilski*, 130 S. Ct., at 3227 *Benson*, 409 U.S., at 71, the *Flook* Court's generalization remains accurate: this Court has never approved the patentability of a computer-implemented process which involved neither special apparatus nor transformation of matter. This Court should now hold, in keeping with its unbroken precedents, that computer software, in particular, cannot be the sole component of a patentable process. To patent a process implemented in computer software, the invention claimed must additionally include either

a special purpose apparatus, not merely a general purpose computer to execute the software, or a transformation of matter.

A. HISTORICALLY, ANY SUBJECT MATTER THAT PRE-EMPTS THE FREE USE OF LAWS OF NATURE, ABSTRACT IDEAS, OR ALGORITHMS IS UNPATENTABLE

Since before the Civil War, this Court has consistently made it clear that subject matter which would have the practical effect of monopolizing laws of nature, abstract ideas or mathematical algorithms is ineligible for patent protection. *O'Reilly* v. *Morse*, 56 U.S. (15 How.) 62, 113 (1854); *Gottschalk* v. *Benson*, 409 U.S. 63; *Parker* v. *Flook*, 437 U.S. 584; *Diamond* v. *Chakrabarty*, 447 U.S. 303; *Diamond* v. *Diehr*, 450 U.S. 175; *Bilski* v. *Kappos*, 130 S. Ct. 3218; *Mayo Collaborative Services* v. *Prometheus Laboratories, Inc.*, 132 S. Ct. 1289.

In *O'Reilly* v. *Morse*, the Court rejected Samuel Morse's claim to the use of "electromagnetism, however developed, for making or printing intelligible characters, signs or letters, at any distances." 56 U.S., at 112 (internal quotation marks omitted). The Court said:

> If this claim can be maintained, it matters not by what process or machinery the result is accomplished. For aught that we now know, some future inventor, in the onward march of science, may discover a mode of writing or printing at a distance by means of the electric or galvanic current, without

> using any part of the process or combination set forth in the plaintiff's specification. His invention may be less complicated — less liable to get out of order — less expensive in construction, and in its operation. But yet, if it is covered by this patent, the inventor could not use it, nor the public have the benefit of it, without the permission of this patentee.

Id. at 113.

In *Benson*, this Court considered the claim to a method for converting numerical information from binary-coded decimal numbers into pure binary numbers, for use in programming conventional general-purpose digital computers. The Court concluded that "[t]he mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly preempt the mathematical formula involved and in practical effect would be a patent on the algorithm itself." 409 U.S., at 71–72. Accordingly the claims were held ineligible under §101.

In *Parker* v. *Flook*, the Court held that to be eligible for patent protection, "[t]he process itself, not merely the mathematical algorithm, must be new and useful." 437 U.S., at 591; *see also*, *Funk Brothers*, 333 U.S., at 130. The Court further stated in *Flook* that it is "incorrect[] [to] assume[] that if a process application implements a principle in some specific fashion, it automatically falls within the patentable subject matter of §101." 437 U.S., at 593. This Court explained that such an assumption is based on an impermis-

sibly narrow interpretation of its precedents, including specifically *Benson*, and is "untenable" because "[i]t would make the determination of patentable subject matter depend simply on the draftsman's art, and would ill serve the principles underlying the prohibition against patents for 'ideas' or phenomena of nature." *Id*.

In alignment with *Benson* and *Flook*, this Court's decision in *Diamond* v. *Diehr* held that structures or processes must, when considered as a whole, perform functions intended to be covered by patent law in order to be eligible for patent protection. 450 U.S., at 192.

In rejecting the patentability of a "business method" implemented in computer software, the Court in *Bilski* stated that "[A]llowing [the claims] would preempt use of this approach in all fields, and would effectively grant a monopoly over an abstract idea." 130 S. Ct., at 3231. The Court also held that such claims cannot be made patent eligible by "limiting an abstract idea to one field of use or adding token postsolution components," thereby affirming the rejection of the claims under §101. *Id*.

More recently, in *Mayo*, while rejecting the claimed processes as "routine, conventional activity previously engaged in by researchers in the field," the Court stated that its decisions

> warn us against interpreting patent statutes in ways that make patent eligibility 'depend simply on the draftsman's art' without reference to the 'principles underlying the prohibition against patents for [natural laws].' *Flook*, 437 U.S., at 593. They warn us against upholding patents that claim processes that too broadly

> preempt the use of natural law. *Morse*, 56 U.S., at 112–20. And they insist that a process that focuses upon the use of a natural law also contain other elements or a combination of elements.

132 S. Ct., at 1294.

Benson, *Flook*, *Diehr*, and the other decisions of this Court regarding patentable subject matter consistently establish that the inquiry into whether subject matter is eligible for patenting is one of substance, not form.

This substantive standard ensures that skilled patent draftsmanship is not capable of overcoming one of the core doctrines of patent law recognized by this Court for more than 150 years: that "[a] principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right." *Le Roy* v. *Tatham*, 55 U.S. (14 How.) 156, 175 (1853).

B. Computer Programs are Algorithms for Computers to Execute Written in Human-Readable Terms. Standing Alone, Without Specialized Machinery or the Transformation of Matter, They Are Not Patentable, As This Court Has Repeatedly Held

This Court has repeatedly addressed the issue whether software is patentable subject matter, and has never found software standing on its own an appropriate subject of patent monopoly, no matter how the claims have been drafted.

In *Microsoft* v. *AT&T*, 550 U.S. 437 (2007), the Court stated that software program code is an idea without physical embodiment and is merely information—a detailed set of instructions. Such abstract ideas without physical embodiment cannot be the subject of a statutory patent monopoly because, "[a]n idea of itself is not patentable." *Rubber-Tip Pencil* v. *Howard*, 87 U.S. (20 Wall.) 498, 507 (1874).

A computer program, no matter what its function, is nothing more or less than a collection of abstract ideas comprising one or more algorithms. It is not conceptually different from a list of steps written down with pencil and paper for execution by a human being. In fact, computer software in source code form is *precisely* a list of steps written for the reading of human beings, who can learn from, and fix errors in, the computer program represented. Further, just as claiming fifty—or even a thousand—laws of nature is no more patentable than claiming a single law of nature, no form of software, regardless how many algorithms or formulas it comprises, is patentable. In no uncertain terms, this Court in *Benson*, 409 U.S., at 71–73, held that software, which contains and upon command executes algorithms that solve mathematical problems through the use of a computer, is not patentable under §101.

Thus, as the Court's precedents unambiguously show, software standing alone, without the presence of a special purpose machine or the act of transforming a particular article into a different state or thing, is merely information, a representation of an algorithm or algorithms, and not a "process" within the meaning of §101.

This Court's decision in *Diamond* v. *Diehr* is not to the contrary. It followed the teaching of *Benson*, ap-

plied in substance the "machine or transformation" test, and determined that the invention before the Court was not substantially the software, but rather the totality of an "industrial process for the molding of rubber products," which was undeniably included within the realm of patentable subject matter. 450 U.S., at 191–93. Had the applicant sought to claim the software used in that process by itself, however, the Court would surely have found it to be unpatentable subject matter just as it had in *Benson*. As the *Diehr* Court observed:

> [W]hen a claim recites a mathematical formula (or scientific principle or phenomenon of nature), an inquiry must be made into whether the claim is seeking patent protection for that formula in the abstract. A mathematical formula as such is not accorded the protection of patent laws, and this principle cannot be circumvented by attempting to limit the use of the formula to a particular technological environment.

450 U.S., at 191 (internal citations omitted).

This result—which makes software describing a portion of the solution to a practical problem unpatentable on its own, outside the real-world context of the problem and its solution—is not only in accord with the rest of this Court's patent jurisprudence, it is also the best way to protect innovation in software, and the only way that fully comports with both Article I, §8 and the First Amendment.

Thus, this Court's precedent repeatedly sets out that software, which is nothing more than a set

of instructions—an algorithm—to be performed by a computer in order to solve some technical or mathematical problem, is subject matter that is not patentable under §101.

C. THE "MACHINE OR TRANSFORMATION" TEST IS THE CORRECT AND COMPLETE TEST OF PATENT ELIGIBILITY FOR COMPUTER-IMPLEMENTED INVENTIONS

This Court held in *Bilski* v. *Kappos* that the "machine or transformation" approach is not the sole determinative measure of the patent eligibility of all processes. 130 S. Ct., at 3227. But the issue in the present case is narrower than that posed to the Court in *Bilski*. The question presented here concerns the patentability of computer software that duplicates the effects of a process previously undertaken without the benefit of computer assistance. No special apparatus or transformation of matter having been presented as part of the claims, the subject matter is unpatentable. In this narrower domain it is appropriate for the Court, in line with its prior decisions, to hold that the "machine or transformation" test is the exclusive test for patent eligibility of computer-implemented inventions.

The Court in *Bilski* said that "there are reasons to doubt" that the "machine or transformation" approach can be the exclusive test for the patentability of "inventions in the Information Age." 130 S. Ct., at 3227. But when the question is narrowed to whether software standing alone should be patentable, there is little reason indeed for doubt. The Court has never so far faced an instance in which the "machine or transformation" test failed to distinguish between patent eligible and ineligible subject matter of this kind. Far

from being a source of uncertainty, as the *Bilski* Court suggested it might be, *Id.*, the "machine or transformation" test would provide substantial certainty now lacking, by reinforcing the teaching of the unbroken precedent of 150 years.

No doubt the pace of change in the area of information technology is rapid. As we show below, the real lesson of contemporary technological development, however, is that patenting has had no positive effect on innovation in software. Uncertainty about what can be patented, on the other hand, has given rise to enormously wasteful litigation. But whatever the pace of innovation, it is unlikely to disclose what has not yet appeared since the beginning of the Information Age: a case in which software standing alone, that fails the "machine or transformation" test, nonetheless is patentable subject matter. In the unlikely event that such a *rara avis* is observed in future, the Court can modify or add to the test. In the meantime, the advantages of certainty that would accrue from the adoption of a clear, bright-line test that cannot be defeated by mere cleverness of draftsmanship would far outweigh the speculative concerns expressed by the Court in *Bilski*.

II. Adhering to the Court's Previous Decisions on Patentability of Software Standing Alone Does Not Imperil the Pace of Software Innovation

The nature of software, like mathematics or basic scientific research, is that innovation is best produced by free sharing. History shows that innovation in software over the last generation has occurred first

in communities of free sharing, where patenting has been systematically discouraged.

A. Innovation in Software, Like Innovation in Mathematics, is Encouraged by Scientific Processes of Free Sharing and Open Publication, Not by Granting State-Issued Monopolies on Ideas

If mathematics were patentable, there would be less mathematical innovation. Only those who were rich enough to pay royalties, or who benefited from subsidization by government, or who were willing to sign over the value of their ideas to someone richer and more powerful than themselves, would be permitted access to the world of abstract mathematical ideas. Theorems build upon theorems, and so the contributions of those who could not pay rent—and all the further improvements based upon those contributions—would be lost.

The principle that innovation is made possible by the free exchange of ideas is not recent, and is not limited to software. Indeed, our constitutional system of free expression since Thomas Jefferson is based on the recognition that control of ideas by power has never produced more ideas than their free and unrestricted circulation. The history of western science since the 17th century is one long testament to this truth, and it is that very history which gave rise to the patent system, whose exclusion of "abstract ideas," "laws of nature," and "algorithms" is as much a recognition of the principle as is the basic constitutional policy of offering temporary legal benefits in return for prompt and complete disclosure of technological discoveries to the public.

B. The History of the Free Software Movement and the Worldwide Adoption of Free and Open Source Software by Industry Shows That Patenting Software Has Not Contributed to the Important Software Innovations of the Last Generation

For more than a quarter century, beginning with a few stalwart thinkers and exponentially increasing in size and influence, a movement to build computer software by sharing—treating software programming languages like mathematical notation, for the expression of abstract ideas to be studied, improved, and shared again—has revolutionized the production of software around the world. The "free software movement," and the developers of "open source software" (collectively described hereinafter as "FLOSS developers") believe, like this Court, that computer software expresses abstract ideas. FLOSS developers therefore conclude that the ideas themselves will grow best if left most free to be learned and improved by all. Their conviction has been shared by hundreds of thousands—soon millions—of programmers around the world, who have devoted their skills to making new and innovative software through the social process that for centuries has been the heart of Western science: "share and share alike."

FLOSS has become the single most influential body of software around the world. In the more than twenty years of its existence, FLOSS has taken the world by storm and has driven the majority of the world's technological advancement in computer programming. FLOSS lives under the hood of it all—from desktops and servers, to laptops, netbooks, smart-

phones, and "the cloud." Linux, distributed under the GNU General Public License of the Free Software Foundation, is the operating system kernel in devices such as mobile phones, networking equipment, medical devices, and other consumer electronics. Android, which relies on Linux and includes the Java programming language and other software under the Apache Software Foundation's ALv2 license, currently has far and away the largest market share in smartphone operating system software. There is no major or minor computer hardware architecture, no class of consumer electronics, no form of network hardware connecting humanity's telephone calls, video streams, or anything else transpiring in the network of networks we call "the Internet" that doesn't make use of FLOSS. The most important innovations in human society during this generation, the World Wide Web and Wikipedia, were based on and are now dominated by free software and the idea of free knowledge sharing it represents.

Given the widespread use and availability of enterprise applications running on GNU/Linux in "the cloud," FLOSS presently provides the infrastructure at the frontier of computing in society. Big Data analytics rely heavily on FLOSS, such as the Hadoop project of the Apache Software Foundation.

The major technologies of the Web, from its beginning, have been embodied in software without patent restrictions. In the early 1990s, CERN, the European Organization for Nuclear Research, committed the Web's fundamental technologies, including initial web-serving and web-browsing programs, to the public domain. The flexibility and sophistication of the Web we use today depends on freely available scripting languages such as Perl and PHP, invented by FLOSS developers who deliberately did not seek patent monop-

olies for them. From 2000, the World Wide Web Consortium (W3C), which advances and standardizes the technology of the Web, has required its recommended technologies in its standards to be available royalty free with respect to all patent claims of the companies and parties participating in standards-making.

This explosion of technical innovation has occurred for two primary reasons. First, the principal rule of free software, the sharing of computer program source code, has allowed young people around the world to learn and to improve their skills by studying and enhancing real software doing real jobs in their own and others' daily lives. Statutory monopolies on ideas expressed in computer programs would have prevented this process from occurring. Second, by creating a "protected commons" for the free exchange of ideas embodied in program source code without rent-seeking by parties holding state-granted monopolies, FLOSS has facilitated cooperative interactions among competing firms. Google, Facebook, Twitter and other information services used by billions of individuals worldwide could not exist without FLOSS and the collaboration it has spawned.

The FLOSS developers and projects that comprise this world-wide movement generally do not own any patents, not only because they have no resources to file for state-granted monopolies, but also because the monopolization of ideas contradicts their fundamental values.

This Court has recognized the growth and innovation in the software industry in the absence of patent protection. In *Benson*, the Court noted that "'the creation of programs has undergone substantial and satisfactory growth in the absence of patent protection and that copyright protection for programs is

presently available.'" 409 U.S., at 72 (quoting "To Promote the Progress of . . . Useful Arts," Report of the President's Commission on the Patent System (1966)). In *Diehr*, the Court subsequently observed that "[n]otwithstanding fervent argument that patent protection is essential for the growth of the software industry, commentators have noted that 'this industry is growing by leaps and bounds without it.'" 450 U.S., at 217 (internal citations omitted).

Mere speculative doubts about the "machine or transformation" test in the "Information Age," *Bilski*, 130 S. Ct., at 3227, must give way to the reality of contemporary information technology. Sharing makes software innovation. Patenting of software standing alone constitutes the monopolization of ideas, which not only violates our constitutional principles but also interferes practically with software innovation. If this Court holds firmly to its prior course, technological innovation in software will continue to flourish. Otherwise we can expect more patent war, less product innovation, and less freedom of thought and invention in software.

III. The First Amendment Prohibits Construing the Patent Act to Permit the Patenting of Abstract Ideas

This Court held in *Eldred*, 537 U.S., at 219, that the First Amendment precludes the extension of statutory monopolies to abstract ideas. As the Court then observed, the near-simultaneous adoption of the Patent and Copyright Clause and the First Amendment indicates that these provisions are fundamentally compatible. *Id.* This compatibility, however, depends on

a construction of the patent and copyright acts that preserves First Amendment principles, including the freedom to communicate any "idea, theory, and fact." *Id.*

Eldred identified two mechanisms in copyright law that are necessary to accommodate this principle. First, the idea/expression dichotomy limits copyright's monopoly to an author's expression, leaving ideas "instantly available for public exploitation." *Id.* Second, the fair use doctrine allows the public to use even copyrighted expression for some purposes, "such as criticism, comment, news reporting, teaching. . . , scholarship, or research." *Id.* at 220.

Patent statutes, which depend on the same constitutional grant of authority as copyright statutes, are similarly limited by the First Amendment. *See id.* at 201 ("Because the Clause empowering Congress to confer copyrights also authorizes patents, congressional practice with respect to patents informs our inquiry"). The presence of an unwavering exemption for abstract ideas reconciles patent law with the First Amendment in a fashion similar to the idea/expression dichotomy's crucial role in reconciling copyright and freedom of speech. The presence of a limiting principle is even more necessary with respect to patent law than with respect to copyright, because, as the Court observed in *Eldred*, "the grant of a patent . . . prevent[s] full use by others of the inventor's knowledge." *Id.* at 217 (internal citation omitted). Patents can and do limit the application of knowledge to produce a new machine or to transform an article into a different state or thing, but they cannot constitutionally limit the communication of knowledge or ideas. *Eldred* teaches that, without this limitation, determining the scope of patent eligibility in each in-

dividual case would raise First Amendment questions of great difficulty.

Patent law also recognizes no analogue to fair use, previously described by this Court as the second bulkwark of constitutional harmony between copyright and free expression. *Id.* at 219–20. The absence of any provision for fair use substantially increases the constitutional difficulty when patents are sought and granted for expressions of abstract ideas.

Without the "machine or transformation" test, dissemination of software standing alone, in source code form, could result in patent infringement. This would fatally disturb the "definitional balance" between the First Amendment and the Patent Act. *Id.* at 219 (quoting *Harper & Row, Publishers* v. *Nation Enterprises*, 471 U.S. 539, 556 (1985)). In its unprocessed source code form, software is merely the expression of abstract ideas in human language—a description of a sequence of steps that will produce a particular result (i.e. an "algorithm"). The source code of a program which performs the steps described in a software patent is distinguishable from the literal patent only in that it expresses the same steps in a different language. Therefore, since anyone may copy or publish the actual patent without infringing, it must also be permissible to communicate its claims in source code form.

The sharing of source code is also essential to "scholarship and comment," two categories of speech recognized in *Eldred*, 537 U.S., at 220, and *Harper &Row*, 471 U.S, at 560, as particular First Amendment concerns. Computer science textbooks, for example, rely heavily on source code and pseudo-code to communicate concepts and describe useful algorithms. *See, e.g.,* Brian W. Kernighan and Dennis M. Ritchie, The C

Programming Language (Prentice Hall 1978). Likewise, computer science students are often required to express their answers to test questions in a real or hypothetical programming language. And without the use of source code, it is difficult for developers to comment on whether an idea can be implemented, to comment on an algorithm's performance, or to suggest improvements.

The "machine or transformation" test serves the purpose of securing accommodation with the First Amendment by ensuring that patent claims on computer-implemented inventions cannot be comprised solely of ideas communicated in computer program code. By requiring a physical special-purpose apparatus or a material transformation, the test implements a construction of §101 that automatically avoids conflict with the First Amendment. If the "machine or transformation" test is *not* the exclusive delimitation of §101 as applied to computer-implemented inventions, what alternative proposal do petitioners and their *amici* advance to avoid First Amendment problems?

CONCLUSION

The "machine or transformation" test evolved over 150 years of this Court's jurisprudence should be affirmed as the necessary criterion for the patenting of inventions implemented in software.

For the foregoing reasons, the decision below should be affirmed.

Respectfully submitted,

EBEN MOGLEN
Counsel of record
MISHI CHOUDHARY
JONATHAN D. BEAN
Software Freedom Law Center
1995 Broadway, 17^{th} Floor
New York, New York 10023
moglen@softwarefreedom.org
212-461-1900

February 27, 2014