Washington and Lee Law Review

Volume 43 | Issue 4 Article 9

Fall 9-1-1986

The Scope Of Copyright Protection For Computer Programs: Exploring The Idea/ Expression Dichotomy*

THE SCOPE OF COPYRIGHT PROTECTION FOR COMPUTER PROGRAMS: EXPLORING THE IDEA/ EXPRESSION DICHOTOMY*

The piracy of computer software[1] is a major problem in the computer industry and costs the industry billions of dollars in lost revenues.[2] Commercial software developers ordinarily invest millions of dollars to create

* A version of this article was submitted in the Nathan Burkan Memorial Competition.

[1] *See* Act of Dec. 12, 1980, Pub. L. No. 96-517, § 10(a), 94 Stat. 3015, 3028 (codified at 17 U.S.C. § 101, 117 (1982)). The 1980 amendment to § 101 of the 1976 Copyright Act defines a computer program as a series of direct or indirect instructions designed for use in a computer to bring about a certain result. *Id.* § 101. Courts often use the term "software" interchangeably with the term "computer program." *See, e.g.*, Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1242 (3d Cir. 1983) (using term "software" interchangeably with term "computer program"), *cert. dismissed*, 464 U.S. 1033 (1984); Centurion Indus., Inc. v. Warren Steurer & Assoc., 665 F.2d 323, 324 (10th Cir. 1981) (describing computer programs created by user to perform desired result as "software"); District of Columbia v. Universal Computer Assoc., 465 F.2d 615, 616 (D.C. Cir. 1972) (referring to computer programs stored on punched cards as software). The term "software," however, also may include the user manuals, program listings and documentation, libraries of program segments known as subroutines, and program specifications as well as the program. BURTON, A DICTIONARY OF MICROCOMPUTING 144 (1976). Software may encompass anything that is not hardware or firmware. *Id.* Furthermore, software often refers to a particular genus of computer programs. For example, systems software is the genus of programs that control the activities of the computer and allows the computer to control and coordinate the function of other hardware. *Id.* Applications software is the genus of computer programs that perform a particular function, like word processing. *Id.*

Hardware refers to the physical machinery including the computer, printer, and various other devices for the output, input, and storage of information. *See* Response of Carolina, Inc. v. Leaseco Response, Inc., 537 F.2d 1307, 1326 (1976) (defining hardware as actual physical machinery that consists of discrete parts including the central processing unit (CPU), electronic device that performs arithmetic and logical operations, and various devices for storing and retrieving data); Triangle Underwriters, Inc. v. Honeywell, Inc., 604 F.2d 737, 739 (2d Cir. 1979) (defining hardware as computer, printer, collator, and related equipment). Firmware describes computer programs stored in Read Only Memory (ROM). THE MCGRAW-HILL COMPUTER HANDBOOK G-4 (H. Helms ed. 1983); *see infra* note 54 and accompanying text (explaining Read Only Memory).

[2] *See Justice Department Okays Promulgation of Industry Standards for Computer Software Protection System*, [Jan.-June] ANTITRUST & TRADE REG. REP. (BNA) No. 5, at A-7 (Jan. 8, 1986) [hereinafter cited as *Industry Standards For Computer Software Protection].* In a business review letter dated January 6, 1986 Assistant Attorney General Douglas Ginsburg stated that unauthorized copying of computer software is a serious economic problem for the computer software industry. *Id. See Thou Shalt Not Dupe*, COMPUTERWORLD, Jan. 28, 1985, at ID/1-ID/2 (software piracy is severe problem in computer software industry). The Association of Data Processing Service, Organizations, Inc. (ADAPSO) recently conducted a study which indicated that the unauthorized copying of computer software has cost the industry $1.3 billion between 1981 and 1984. *Industry Standards For Computer Software Protection, supra*, at A-7. Both forms of unauthorized copying, piracy and illicit copying, are serious problems in the computer industry. BORKING, THIRD PARTY PROTECTION OF SOFTWARE AND FIRMWARE 408-13 (1985). Piracy is the commercial exploitation of another's software while illicit copying involves the unauthorized copying of computer software for personal use. *Id.* Both forms of unauthorized copying infringe the exclusive rights of the copyright owner to make copies. *See* 17 U.S.C. § 106 (1982) (copyright owner has exclusive right to make copies of copyrighted works).

and market successful new computer programs.[3] Accordingly, legal protection of software is essential for the computer industry to thwart software piracy and insure software developers' monetary incentives.[4] Many software developers rely on copyright protection as the most accessible form of legal protection because of the limited availability of other forms of legal protection for software.[5] The extent to which copyright law protects more than the

[3] *See Software: The New Driving Force,* Bus. Wk., Feb. 27, 1984, at 77. The estimated average cost of developing and marketing a new microcomputer program is eight million dollars. *Id.* Presumably, the cost of developing a longer, more complex program for a mainframe computer is in excess of eight million dollars. Brief of Amicus Curiae at 3, Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986).

[4] *See* Apple Computer, Inc. v. Formula Int'l, Inc., 562 F. Supp. 775, 783 (C.D. Cal. 1983), *aff'd,* 725 F.2d 521 (9th Cir. 1984). Few companies will invest the time and capital necessary to develop new productivity-enhancing programs if other companies can appropriate their products. *Id.; see* Final Report Of The National Commission On New Technological Uses Of Copyrighted Works 11 (1978) (legal and physical protection of computer programs is necessary for future creation and dissemination of programs) [hereinafter cited as CONTU Final Report].

[5] *See* CONTU Final Report, *supra* note 4, at 16-18 (comparing copyright with other forms of legal protection for computer software). Several forms of legal protection are available to the developers of computer software, including copyright protection, patent protection, trade secrecy protection, and protection from unfair competition. *Id.* at 16-19.

Patents provide a monopoly to inventors for any "new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement." *See* 35 U.S.C. § 101 (1982). A valid patent gives the owner the exclusive right to use, make, or sell the patented invention. *See* 35 U.S.C. §§ 154, 271(a) (1982). *See generally* Conley and Bryan, *A Unifying Theory for the Litigation of Computer Software Copyright Cases,* 63 N.C.L. Rev. 563, 569 (1985) (discussing patent protection for computer software); Stout, *Protection of Programming in the Aftermath of Diamond v. Diehr,* Computer/Law Journal, Vol. IV, 207, 208 (1983) (same). Although the Supreme Court in *Diamond v. Diehr* held that a computer program may be the proper subject matter for patent protection in limited circumstances, patents are difficult to obtain for computer programs. *See* International Bureau of the World Intellectual Property Organization, Model Provisions on the Protection of Computer Software 4 (1978) (discussing difficulty of obtaining patents for computer programs); *see* Diamond v. Diehr, 450 U.S. 175, 185 (1981). Obtaining a patent can take at least two years and can cost several thousand dollars. Stout, *supra,* at 208. Patent protection is, therefore, not feasible for some programs because the commercial life-cycle of some programs is shorter than the time necessary to obtain the patent. A. Gemignani, Law and the Computer 100, 107 (1981).

Although trade secrecy is one form of protection relied upon by the computer industry, problems exist with trade secrecy protection, especially with mass marketed software. Gemignani, *supra,* at 113. Trade secrecy protects formulas, patterns, devices, or compilations of information that a business uses to gain an advantage over competitors who do not have the secret. Restatement of Torts § 757 comment b (1939). The Restatement of Torts suggests several elements to consider when deciding whether certain information is a trade secret. *Id.* § 757. These considerations include the extent to which persons outside the business know the secret, whether employees of the business know the secret, the degree of care the business has taken to protect the secret, the value of the secret to the business and its competitors, the cost of developing the secret, and the ease or difficulty with which others can duplicate or acquire the secret. *Id.* To meet the secrecy requirement of a trade secret, a business does not have to maintain absolute secrecy, but can disclose to others in confidence under an express or implied agreement not to disclose the agreement. Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 475 (1974). An obvious tension exists between maintaining secrecy and marketing programs for wide literal instructions (code) of a program, however, is uncertain, and software developers presently cannot rely on copyright protection alone.[6] Greater certainty in the scope of copyright protection for computer programs is necessary to promote the continued growth of the software industry.[7]

Much uncertainty surrounding the legal protection for computer programs is a result of the fundamental copyright principle that copyright law protects only the expression of an idea, and not the underlying idea itself.[8]

distribution. Conley & Bryan, *supra,* at 576. Nevertheless, the wide distribution of software alone does not prohibit trade secrecy protection. *Id.* The developer, however, can preserve the trade secrecy by distributing the programs pursuant to a nondisclosure agreement that creates a confidential relationship between the developer and the licensee. *Id.; see* J & K Computer System, Inc. v. Parrish, 642 P.2d 732, 735 (Utah 1982) (finding that developer had preserved trade secrecy protection by informing employees and customers that program was secret, and by placing notice on each copy of program restricting program's use to authorized licensees). In addition to the difficulty in maintaining the secrecy of widely distributed software, uncertainty exists concerning the extent of trade secrecy protection and the remedies available for misappropriation. *See* Conley & Bryan, *supra,* at 576-78 (discussing problems with protection of computer software through trade secrecy); GEMIGNANI, *supra,* at 113 (same). For example, trade secrecy does not offer protection against independent creation or reverse engineering—the method of starting with a particular result and working backwards to ascertain the process that created the result. *Kewanee,* 470 U.S. at 476.

An alternative form of protection for software is the common-law doctrine of unfair competition. *See* CONTU FINAL REPORT, *supra* note 4, at 18. Unfair competition protects against the misappropriation of a competitor's skill, investments, and labor. *Id.* The unfair competition doctrine prohibits the counterfeiting or "passing off" of another's work. *Id.* In certain situations, the doctrine of unfair competition can provide ancillary support to copyright protection. *Id.* The doctrine, however, has a limited scope of protection and probably cannot provide sufficient protection alone for computer programs. *Id.*

Finally, many software developers are turning to contract law to protect their rights in software through license agreements. *See* Note, *The Protection of Computer Software Through Shrink-Wrap License Agreements,* 42 WASH. & LEE L. REV. 1347, 1349-50 (1985) (by using license agreements developer can limit users' rights to copy, transfer, and modify software and thereby acquire greater legal protection than by using traditional methods alone).

[6] *See* Note, *supra* note 5, at 1356 (scope of protection provided by copyright is uncertain and industry cannot rely on copyright alone to prevent piracy). Many software developers do not trust the protection provided by copyright law and apparently cannot depend on copyright law to protect their computer programs because of unresolved issues regarding protection of computer programs. BORKING, *supra* note 2, at 253; *see* Ranney, *Copyright Law Could Threaten Software Mimics,* INFOWORLD, Dec. 2, 1985, at 1, 8 (software developers do not know how closely one program can resemble another program without infringing copyright); *infra* notes 7-10 and accompanying text (discussing uncertainty in copyright protection of computer software).

[7] *See* RANNEY, *supra* note 6, at 8 (software developers need greater certainty in scope of protection provided by copyright law); *supra* note 4 and accompanying text (discussing need for software protection to provide software developers with sufficient monetary incentives to create new software).

[8] *See* 17 U.S.C. § 102(b) (1982). Copyright law does not protect any idea, process, or method of operation. *Id.; see* H.R. REP. No. 1476, 94th Cong., 2d Sess. 57 (1976), *reprinted in* 1976 U.S. CODE CONG. & AD. NEWS 5659, 5670 (section 102(b) provides that only expression of idea is copyrightable) [hereinafter cited as 1976 REPORT]; *infra* note 24 and accompanying text (discussing fundamental copyright principle that copyright law protects expression of idea only).

The boundary between protectible expression and the underlying idea or process in computer programs is particularly difficult to determine.[9] In distinguishing between the idea and the expression in a computer program, courts and commentators have recognized that limiting copyright protection only to the programming code in a computer program will not insure proper monetary incentives for program developers.[10] Consequently, courts have begun to protect additional elements in computer programs such as the structure and organization of programs.[11] Although courts recently have begun to address the issue of protectible expression in computer programs, the scope of protection for computer programs remains undetermined.[12]

The policies underlying the federal copyright law are instructive for determining the proper scope of protection for computer programs. The copyright clause of the United States Constitution authorizes Congress to protect the "writings" of "authors" to promote the advancement of science

[9] CONTU Final Report, *supra* note 4, at 22; *see* Universal Athletic Sales Co. v. Salkeld, 511 F.2d 904, 907 (3d Cir.) (no principle can determine when imitator has gone beyond 'idea' and borrowed 'expression'), *cert. denied*, 423 U.S. 863 (1975); *infra* notes 46-50 and accompanying text (discussing difficulties in distinguishing idea from expression in computer software).

[10] *See* Apple Computer, Inc. v. Formula Int'l, Inc., 562 F. Supp. 775, 783 (C.D. Cal. 1983), *aff'd*, 725 F.2d 521 (9th Cir. 1984) (software companies will not invest capital and resources if people can freely duplicate their programs). Because the coding level of creating a program only represents 20% of a new program's total development cost, few companies will invest the time and resources necessary to create new programs when their competitors freely can pirate eighty percent of the investment. *See* Brief of Amicus Curiae, at 6, Whelan Assoc. Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986) (limiting copyright protection to literal text of computer programs would seriously harm software industry and impair software industry's contributions to economy and nation); Frank, Critical Issues in Software, 20-24 (1983) (actual coding represents only 20% of development cost of software). If legal protection of computer software is inadequate, those companies that did decide to develop new programs would have to charge exorbitant prices to recoup their investments. *See* CONTU Final Report, *supra* note 4, at 11 (without adequate protection of software, developers would have to recoup their investment by charging exorbitant prices on first sales of software).

The most important intellectual labors of a programmer are in creating the design, structure, and organization of a program. Brief of Amicus Curiae, at 10, Whelan Assoc. Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986). A programmer creates works of authorship at every stage in the development of a computer program. *Id.* The programmer constantly refines each step into successively more detailed steps. *Id.* at 9. The final stage of the program, translating a detailed expression or description of the steps necessary to achieve the desired result into source code, does not require as much creativity as the earlier stages of software development because the coder must preserve the expression embodied in the program's development stages. *Id.* at 10. Protection of the authorship involved in the earlier stages of programming, therefore, is necessary to insure adequate incentives for computer programmers. *Id.* at 13.

[11] *See infra* notes 77-140, 164-67 and accompanying text (discussing cases that extend copyright protection of computer software to structure and organization of program).

[12] *See infra* notes 77-140, 164-67 and accompanying text (discussing cases that extend copyright protection to computer program's structure and organization).

and the arts.[13] To achieve the objective of promoting science and the arts, the United States Constitution gives Congress the power to reward authors through exclusive control over the right to make and sell copies of the authors' works, thereby encouraging people to dedicate themselves to intellectual and artistic creation.[14] Copyright law, consequently, serves two inherently competing interests: rewarding individual's creative efforts as an incentive for creating useful programs and protecting the free dissemination of ideas for the public's benefit.[15]

Historically, federal copyright law has expanded to protect novel forms of expression made possible by the creation of new technologies.[16] By 1976,

[13] U.S. CONST. art. I, § 8, cl. 8. The copyright clause states in pertinent part:

> The Congress shall have Power . . . [8] To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries;

Id. The objective of the copyright clause is to promote the advancement of science and art. *See* Goldstein v. California, 412 U.S. 546, 555 (1973). In *Goldstein,* the Supreme Court defined the constitutional meaning of "author" as the person to whom a work owes its origin. *Id.* The Supreme Court defined the constitutional meaning of "writing" to include "any physical rendering of the fruits of creative intellectual or aesthetic labor." *Id.*

[14] U.S. CONST. art. I, § 8, cl. 8; *see Goldstein v. California,* 412 U.S. 546, 555 (1973) (Constitution gives Congress power to reward authors and inventors by granting them control over sale or commercial use of copies of their works).

[15] *See* Whelan Assoc., Inc v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1235 (3d Cir. 1986) (purpose of copyright law is to create most productive balance between incentive and dissemination of information); Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corporation, 562 F.2d 1157, 1163 (9th Cir. 1977) (copyright law attempts to reconcile two competing social interests: rewarding individuals creativity and effort while permitting nation to benefit from use of same subject matter).

[16] *Id.* at 562. Congress historically has decided which works of authorship to protect under federal copyright law by considering both the character of the "writing" and the commercial importance of the product to the national economy. *Id.* In 1790 Congress passed the Act of May 31, 1790, the first copyright statute, which protected maps, charts, and books. Act of May 31, 1790, c. 15, 1 Stat. 124. Since 1790, however, Congress has expanded copyright law to protect the many new forms of expression. *See* Act of Apr. 29, 1802, c. 36, 2 Stat. 171. Congress amended the 1790 Act in 1802 to protect any historical or other prints. *Id.* Congress extended protection to musical compositions in 1831. Act of Feb. 3, 1831, c. 16, 4 Stat. 436. When Matthew Brady's pictures of the Civil War began attaining fame in 1865, Congress added protection for photographs and photographic negatives. Act of Mar. 3, 1865, c. 126, 13 Stat. 540. In 1870 Congress extended protection for paintings, drawings, chromos, statuettes, statuary, and models or designs of fine art. Act of July 8, 1870, c. 230, 16 Stat. 198. *See also Goldstein,* 412 U.S. at 562 (discussing history of copyright statutes). In response to the enormous increase in various kinds of expression needing protection under federal copyright law, Congress passed the Copyright Act of 1909, which consolidated and amended the existing federal copyright statute. Copyright Act of 1909 (1909 Act), Pub. L. No. 60-349, 35 Stat. 1075. *See* H.R. REP. No. 2222, 60th Cong., 2d Sess. 1 (1909) (enormous increase in copying of works of authorship not subject to prior protection under prior copyright acts necessitated revision in copyright law). Since 1909, the creation of new technologies has provided authors with novel forms of expressing themselves that often have created difficult issues about the protection of these new works of authorship. Copyright law has struggled to accommodate the new forms of expression. *See* CONTU FINAL REPORT, *supra* note 4, at 3 (noting that some people have questioned adequacy of legal structure to cope with pace and technological change in copyrightable works). In 1912 Congress expanded the 1909 Act to include specifically motion pictures, reasoning that the property rights and money invested in motion pictures had become so valuable that copyright

the tremendous growth in technology required a major revision of the copyright laws,[17] and Congress enacted the Copyright Act of 1976[18] ('76 Act). The '76 Act protects "original works of authorship fixed in any tangible medium of expression, now known or later developed. . . ."[19] The '76 Act retained two basic requirements of copyright protection: originality and fixation in a tangible form.[20] In the '76 Act, Congress expanded the fixation requirement to encompass works fixed in any medium capable of communicating the work either directly or through the assistance of a machine.[21] The '76 Act enumerates two categories of copyrightable works of authorship that have been applied respectively to computer programs and their visual screens: literary works; and motion pictures and other audiovisual works.[22]

In section 102(b) of the '76 Act, Congress codified the fundamental copyright principle established by the Supreme Court in *Baker v. Selden,*[23]

law should protect motion pictures. Act of August 24, 1912, c. 566, 37 Stat. 488. The increase in the piracy of sound recordings, which was partially due to technological advances in recording equipment, forced Congress in 1971 to amend the 1909 Act to specifically include sound recordings. Act of Oct. 15, 1971, Pub. L. No. 92-140, 85 Stat. 391.

[17] *See* CONTU FINAL REPORT, *supra* note 4, at 3. The technology explosion left Congress and those affected by the copyright laws dissatisfied with the copyright law under the Copyright Act of 1909. *Id.* The general belief was that a major revision of the copyright law, rather than piecemeal amendments to the 1909 Act, was necessary. *Id.;* KITCH & PERLMAN, LEGAL REGULATION OF THE COMPETITIVE PROCESS 622-23 (1979) ('76 Act altered fundamental structure of American copyright law).

[18] Copyright Act of 1976 ('76 Act) §§ 101-810, 17 U.S.C. §§ 101-810 (1982).

[19] 17 U.S.C. § 102(a) (1982).

[20] *Id.*

[21] *Id; see* 1976 REPORT, *supra* note 8, at 5665 (overruling White-Smith Publishing Co. v. Apollo Co., 209 U.S. 1 (1908)). The United States Supreme Court in *White-Smith Publishing Co. v. Apollo Co.* held that the piano roll for a player piano was not a copy of the musical composition embedded on the roll because the music was not in a form that humans could see and read. *White-Smith Publishing Co.*, 209 U.S. at 18. In contrast, the '76 Act does not require that a work be directly perceivable by humans. 17 U.S.C. § 102(a) (1982). The '76 Act only requires that a work be "fixed" in a medium that can reproduce the work directly or with the assistance of a device or machine. *Id.* For example, a program in the volatile memory of a computer that disappears when the programmer turns off the computer is not "fixed" within the meaning of the '76 Act. *See* 1976 REPORT, *supra* note 8, at 5666 (term "fixed" excludes reproductions momentarily stored in "memory" of computer). If, however, the programmer stores the program onto an electromagnetic tape or disk, which is capable of reproducing the program at another time, the program is fixed within the meaning of the '76 Act. *See id.* (program stored on disk or tape is "fixed" because program is capable of being reproduced with aid of computer).

[22] 17 U.S.C. § 102(a)(1), (6) (1982); *see* 1976 REPORT, *supra* note 8, at 5666. The categories of works of authorship are not intended to be exclusive. 1976 REPORT, *supra* note 8, at 5666. By using the word "include" in § 102(a) of the '76 Act, Congress intended that the categories of works of authorship should be "illustrative and not limitative." *Id.* Computer programs are literary works under existing copyright law. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983) (computer program in source code or object code is literary work), *cert. dismissed,* 464 U.S. 1033 (1984). The term "literary works" does not suggest any standards of literary merit and includes computer programs to the extent that computer programs include the programmer's expression of ideas. 1976 REPORT, *supra* note 8, at 5666. Visual screen outputs are considered audio-visual works under copyright laws. Whelan Assoc. Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1244 (3d Cir. 1986); Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 874 (3d Cir. 1982); 17 U.S.C. § 102(a)(6) (1982).

[23] 101 U.S. 99 (1879).

in which the Court distinguished between an idea and the expression of an idea, and allowed copyright protection for the expression of an idea only.[24] Section 102(b), therefore, denies copyright protection for any "idea, procedure, process, system, method of operation, concept, principle, or discovery."[25] Although the '76 Act implicitly recognized that computer programs were copyrightable works,[26] Congress did not specifically include computer programs in the '76 Act.[27] Instead, Congress deferred legislative action until

[24] 17 U.S.C. § 102(b) (1982); *see* Baker v. Selden, 101 U.S. 99 (1879). In *Baker v. Selden,* the Supreme Court established the fundamental principle that copyright protects only the expression of an idea, and not the idea itself. *Baker,* 101 U.S. at 101-02. In *Baker,* the plaintiff, Selden, owned the copyright in a book describing an accounting system. *Id.* at 102. The book contained blank forms and a description of how to use the forms as a bookkeeping system. *Id.* The defendant, Baker, subsequently published books that used the bookkeeping system described in Selden's book. *Id.* at 107. The Supreme Court held that Baker had not infringed Selden's copyrights in the book because Baker had used only the system contained in Selden's book. *Id.* at 107. The Supreme Court reasoned that Selden had acquired exclusive rights only in the printing and publishing of the book and not in the use of any ideas contained in the book. *Id.* at 102-03.

The Supreme Court expanded the *Baker v. Selden* principle in *Mazer v. Stein. See* Mazer v. Stein, 347 U.S. 201, 217 (1954). In *Mazer,* the Supreme Court considered whether statuettes used as lampbases were the proper subject of copyright protection as "works of art." *Id.* at 204-05. In *Mazer,* the plaintiff, Stein, created statuettes and sold the statuettes both as lampbases and statuettes. *Id.* at 203. Stein owned a registered copyright in the statuettes as "works of art." *Id.* at 202-03. Subsequently, the defendant Mazer, without authorization, copied the statuette and began selling the lamps, using the statuettes as lampbases. *Id.* Stein sued Mazer for infringement of his copyright in the statuettes as "works of art." Id. at 204. Mazer argued that Stein's intended use of the statuettes in the form of a lamp precluded copyright of the statuettes because the statuettes would qualify for a design patent. *Id.* at 215. The Supreme Court held that because the statuettes would qualify for a patent did not preclude the copyrightability of the statuettes as works of art. *Id.* at 217. In distinguishing between patent and copyright, the Supreme Court held that copyright protects only the expression of the idea, not the idea itself. *Id.* The copyright in the statuettes gave Stein no exclusive rights to the use of any statuettes in table lamps, only the exclusive right to prevent others from making lamps that incorporated copies of Stein's statuettes. *Id.* The Supreme Court held that Mazer's use of Stein's statuettes violated Stein's copyright in the statuettes. *Id.*

[25] 17 U.S.C. § 102(b) (1982); *see* 1976 REPORT, *supra* note 8, at 5670. The 1976 Report makes clear that § 102(b) merely codifies the existing judicial doctrine distinguishing between expression and idea in the copyright law. *Id.*

[26] 1976 REPORT, *supra* note 8, at 5667 (definition of "literary works" in legislative history of '76 Act includes computer programs to extent programs incorporate programmer's expression of ideas).

[27] *See* 17 U.S.C. § 117 (1976). Section 117 of the '76 Act, as originally enacted, expressly disclaimed the application of the '76 Act to computer software. *Id.* Congress' intent in maintaining the status quo of copyright protection for computer software was to allow for an interim period to study the problems associated with the copyright of software. *See* 1976 REPORT, *supra* note 8, at 5664. The legislative history of the '76 Act, however, demonstrates that Congress considered computer software the proper subject of copyright protection. *Id.* Congress regarded computer programs as extensions of copyrightable subject matter. *Id.* In response to Congress' concerns that copyright protection might extend protection to processes or methodologies used by a programmer, the 1976 Report clarifies that the expression adopted by the computer programmer and not the actual processes or methods embodied in the program is the copyrightable element in a computer program. *Id.* at 5667, 5670.

the National Commission on New Technological Uses of Copyrighted Works (CONTU), the commission assigned to study the problems related to copyright protection of computer software, had made its recommendations to Congress.[28]

In the 1980 amendments to the '76 Act,[29] Congress adopted the pertinent recommendations contained in *CONTU's Final Report*.[30] The 1980 amendments expressly recognize that computer programs are copyrightable works of authorship.[31] The 1980 amendment to section 101 of the '76 Act defines a computer program as a group of instructions or statements written for use by a computer, either directly or indirectly, to accomplish a particular result.[32]

[28] *See* Pub. L. No. 93-573, 88 Stat. 1873 (1974) (codified at 17 U.S.C. § 201 (1982)). Congress established the National Commission on New Technology Uses of Copyrighted Works (CONTU) and charged CONTU with responsibility for conducting a three year study of the use and reproduction of copyrighted works in connection with computers and the problems associated with machine duplication systems. *Id.*

[29] Act of Dec. 12, 1980, Pub. L. No. 96-517, § 10(a), 94 Stat. 3015, 3028 (codified at 17 U.S.C. § 101, 117 (1982)).

[30] *See* H.R. Rep. No. 1307, 96th Cong., 2d Sess. 23, *reprinted in* 1980 U.S. Code Cong. & Ad. News 6460, 6482 [hereinafter cited as 1980 Report]. The House Report accompanying the passage of the 1980 amendments notes that the revisions contained in the 1980 amendments embody the recommendations made by CONTU and clarify the application of copyright law to computer software. *See* CONTU Final Report, *supra* note 4, at 1. The Final Report recommended amending the copyright law to clarify that computer programs are the proper subject of copyright law whenever a program embodies the author's original creation. *Id.* CONTU also recommended that § 117 ensure that the "rightful possessors" of programs may adapt or copy the program when necessary to use the program or when necessary to make a backup copy for archival purposes. *Id.* Congress codified CONTU's recommendation to allow the "owner of a copy of a computer program" rather than the "rightful possessor" to make copies of programs under the conditions recommended by CONTU. 17 U.S.C. § 117 (1982); *see infra* note 32 (quoting § 117 of '76 Act).

[31] *See* 17 U.S.C. § 101 (1982) (defining computer program).

[32] 17 U.S.C. § 101 (1982). The 1980 amendment added the following definition of "computer program" to the '76 Act:

> A "computer program" is a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.

Id. The 1980 amendment to section 117 provides necessary exceptions to the creator's exclusive right to make copies of their copyrighted works. *See* 17 U.S.C. § 117 (1982). Section 117 allows the owner of a copy of a computer program to make spare copies of the program for archival purposes. *Id.* Section 117 provides in pertinent part:

> . . . [I]t is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation for that computer program provided:
>
> (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner, or
>
> (2) that such a new copy or adaptation is for archival purposes only and that all archival copies are destroyed in the event that continued possession of the computer program should cease to be rightful.

Id.

The owner of a copy of a copyrighted program does not receive necessarily any rights to the copyright by owning a copy of the work because the '76 Act distinguishes between the

Although the 1980 amendment to section 101 specifically recognizes that computer programs are protectible works of authorship, the amendment does not attempt to delineate the exact point between the unprotectible idea and the protectible expression of a computer program.[33] Congress left this difficult and important determination for the federal judiciary to decide on a case-by-case basis.[34]

To distinguish between the idea and expression in computer software, courts have focused on the merger principle of idea and expression.[35] Idea and expression merge when few or no other ways of expressing a particular idea exist.[36] Under the idea-expression merger principle, when the idea and the expression in a computer program become inseparable, copyright law no longer protects the program because copyright protection of the program would give the copyright owner a monopoly on the program's idea.[37] A particular computer program, therefore, is the expression of the idea and copyrightable when other programs can be written that achieve the same

ownership of a copy of a program and the ownership of the copyright itself. *See* 17 U.S.C. § 202 (1982) (ownership of exclusive rights under a copyright is distinct from ownership of material that embodies copyrighted work).

[33] *See* 1980 REPORT, *supra* note 30, at 6462. The House Report does not attempt to define protectible expression in computer software. *Id.*

[34] *See* CONTU FINAL REPORT, *supra* note 4, at 22-23. To the extent that the CONTU Final Report is a source of legislative history, the legislative history of the 1980 amendments suggest that the federal judiciary should delineate the proper scope of copyright protection for computer programs. *Id.; see* Micro-Sparc, Inc. v. Amtype Corp., 592 F. Supp. 33, 35 n.7 (D. Mass. 1984) (CONTU Report comprises entire legislative history for amendment to § 117 of '76 Act); Midway Mfg. Co. v. Strohon, 564 F. Supp. 741, 750 (N.D. Ill. 1983) (CONTU Report reflects congressional intent). *But see* Whelan Assoc. Inc., v. Jaslow Dental Laboratory, 797 F.2d 1222, 1241 (3d Cir. 1986) (CONTU Report cannot substitute for legislative history of § 102(b) of '76 Act). CONTU's Final Report acknowledges the difficulty in separating the protectible element of expression in computer programs and the idea that underlies the expression. *See* CONTU FINAL REPORT, *supra* note 4, at 18 (distinction between copyrightable programs and underlying processes or ideas does not always "shimmer with clarity"). Rather than attempt to differentiate beween the expression and the idea in computer programs, CONTU suggested that the federal judiciary make this determination on a case-by-case basis. *Id.* at 23.

[35] *See, e.g.,* M. Kramer Mfg. Co. v. Andrews, 783 F.2d 421, 436 (4th Cir. 1986) (applying merger of expression and idea test to determine whether program constitutes protectible expression) (citing Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983), *cert. dismissed,* 464 U.S. 1033 (1984)); Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521, 525 (9th Cir. 1984) (inquiry for identifying idea and expression in computer software is no different from inquiry for determining whether idea and expression have merged); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485. 1502 (D. Minn. 1985) (same); *see also* Morrissey v. Procter & Gamble Co., 379 F.2d 675, 678-79 (1st Cir. 1967) (when few, if any, ways exist for expressing particular idea, then idea and expression have merged).

[36] *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d at 1253 (idea and expression merge if other ways of expressing idea are foreclosed as practical matter), *cert. dismissed,* 104 S. Ct. 690 (1984). Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971) (same); *see also supra* note 35 (discussing merger of idea and expression).

[37] *See* M. Kramer Mfg. Co. v. Andrews, 783 F.2d 421, 436 (4th Cir. 1986) (when only one way to express idea exists, then idea and expression merge, and work is not copyrightable); *see also* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983), *cert. dismissed,* 464 U.S. 1033 (1984) (same); CONTU FINAL REPORT, *supra* note 4, at 20 (when no other expression is available to achieve a particular result, programmer can use copyrighted language without infringing copyrighted work).

result as the particular program.[38] The idea in a computer program is the function or result that the program achieves, like a word processing program.[39] Thus, if programmers can create other word processing programs using a different expression that achieves a result similar to a particular word processing program, then the particular word processing program is the expression of an idea and is copyrightable.[40] If other programmers, however, cannot use a different expression to achieve the particular result, then the program is not copyrightable because idea and expression have merged.[41]

Distinguishing between protectible expression and unprotectible ideas or processes in computer programs is essential in cases of copyright infringement because the copyright law protects only an author's expression.[42] To prevail on a claim of copyright infringement, a plaintiff must establish ownership of a valid copyright and show that the defendant impermissibly copied the program.[43] Unless a defendant admits to copying the plaintiff's work or the contested work is an identical copy, plaintiffs have difficulty proving copyright infringement because direct evidence of copying is rare.[44] Courts, therefore, infer copying from a showing that the defendant had access to

[38] *See* Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1238-39 (3d Cir. 1986) (concluding that detailed structure of program was part of program's expression when other programs exist that contain different structure but perform same function); Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521, 525 (9th Cir. 1984) (if programmers can write other programs that perform same function as particular program, then program is expression of idea); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983) (same), *cert. dismissed*, 464 U.S. 1033 (1984); M. Kramer Mfg. Co. v. Andrews, 783 F.2d 421, 436 (4th Cir. 1986) (same). In *M. Kramer Mfg. Co., Inc. v. Andrews*, the Fourth Circuit stated that in the computer area the Third Circuit in *Apple v. Franklin* has enunciated the acceptable test for distinguishing between idea and expression; the test is whether programmers can write other programs that perform same function. *Kramer*, 783 F.2d at 436; *see Johnson*, 623 F. Supp. at 1501-02 (noting the Third and Ninth Circuits have framed idea-expression dichotomy test in terms of single inquiry whether programmers can create other programs that perform same function as copyrighted program).

[39] *See Whelan*, 797 F.2d at 1236 (purpose or function of utilitarian work is work's idea); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983) (function performed by program is idea, not expression, of program), *cert. dismissed*, 464 U.S. 1033 (1984).

[40] *See id.* (noting that if programmers independently are capable of creating other programs that perform same function as particular program, then particular program is expression of idea).

[41] *See supra* note 35 and accompanying text (discussing point at which idea and expression of idea merge).

[42] *See, e.g.*, Sid & Marty Krofft Television v. McDonald's Corp., 562 F.2d 1157, 1163 (9th Cir. 1977) (no infringement occurs if defendant does not copy protected expression in plaintiff's work); Williams Elec., Inc. v. Bally Mfg. Corp., 568 F. Supp. 1274, 1282 (N.D. Ill. 1983) (no copyright infringement exists unless defendant copied protected expression in plaintiff's game); *supra* notes 35-41 and accompanying text (discussing fundamental principle of copyright law that copyright protects only expression of idea, and not idea itself).

[43] *See* Marcelo Ramos Motta, Inc. v. Samuel Weiser, Inc., 768 F.2d 481, 483 (1st Cir.) (to establish copyright infringement plaintiff must prove ownership of valid copyright and "copying" by defendant, *cert. denied*, 106 S. Ct. 596 (1985)); Atari, Inc. v. North Am. Philips Consumer Elec. Corp., 672 F.2d 607, 620 (7th Cir.), *cert. denied*, 459 U.S. 880 (1982) (same); *see also* 17 U.S.C. § 501(a) (1982) (person that violates exclusive rights of copyright owner is infringing copyright).

[44] Atari, Inc. v. Amusement World, Inc., 547 F. Supp. 222, 225 (D. Md. 1981).

the plaintiff's work and that the defendant's work is "substantially similar" to the plaintiff's copyrighted work.[45]

Although courts have established the idea-expression merger test for determining when a computer program constitutes protectible expression under copyright law, the close relationship between expression and process in computer programs makes distinguishing the copying of expression from the underlying process or idea difficult to determine.[46] Because com-

[45] *See* Atari, Inc. v. North Am. Philips Consumer Elec. Corp., 672 F.2d 607, 614 (7th Cir.) (allowing circumstantial proof of copying by showing defendant's access to copyrighted work and "substantial similarity" between works), *cert. denied,* 459 U.S. 880 (1982); Warner Brothers, Inc. v. American Broadcasting, Inc., 654 F.2d 204, 207 (2d Cir. 1981) (same); Williams Elec., Inc. v. Bally Mfg. Corp., 568 F. Supp. 1274, 1281 (N.D. Ill. 1983) (same). Courts infer access to copyrighted works when a copyrighted work has been widely disseminated works. *Atari,* 672 F.2d at 614; *see* Warner Bros. v. American Broadcasting Co., 654 F.2d 204, 208 (2d Cir. 1981). Absent a showing of access, courts require the plaintiff to show that the contested works are so "strikingly similar" that independent creation of the two programs was not possible. Ferguson v. National Broadcasting Co., 584 F.2d 111, 113 (5th Cir. 1978). The general test for determining substantial similarity between two works is whether the average lay observer would recognize contested work as being an appropriation of copyrighted work. *Warner Bros.,* 654 F.2d at 208. The test of substantial similarity is whether ordinary observer would recognize contested work as having been copied from copyrighted source. Wihtol v. Crow, 309 F.2d 777, 780 (8th Cir. 1962); *see also* Arnstein v. Porter, 154 F.2d 464, 468-69 (2d Cir. 1946) (establishing bifurcated substantial similarity test in which expert testimony is relevant for question of copying, but not for question of unlawful appropriation); Sid & Marty Krofft Television Prod., Inc. v. McDonald's Corp., 562 F.2d 1157, 1164-65 (9th Cir. 1977) (bifurcated substantial similarity test involves "extrinsic test" to determine similarity of ideas and "intrinsic test" to determine similarity of expression). Copyright infringement does not require slavishly detailed copying. Comptone Co. v. Rayex Corp., 251 F.2d 487, 488 (2d Cir. 1958). When the accused work captures the "total concept and feel" of the copyrighted work, then the accused work is substantially similar to copyrighted work. *Atari,* 672 F.2d at 614. Substantial similarity does not require absolute identity, rather, courts will examine both the quantitative and qualitative significance of copied matter. *See* In the Matter of Certain Personal Computers and Components Thereof, Investigation No. 337-TA-140 (Final Decision, March 1984) (finding copying to be qualitatively significant when accused work contained 23 of 32 most useful subroutines out of approximately 70 subroutines in copyrighted program).

Because computer programs are stored on disks or embedded into microchips and are not visible, application of the ordinary observer test is problematic in computer context. E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1493 (D. Minn. 1985). Thus, courts have adopted in form, if not by name, an iterative approach to the substantial similarity test. Id. at 1493 (D.C. Minn. 1985); Note, Copyright Infringement of Computer Programs: A Modification of the Substantial Similarity Test, 64 MINN. L. REV. 1264, 1294-300 (1984). The iterative approach requires the plaintiff to prove that the defendant "used" the copyrighted work in developing the defendant's alleged copy and that the defendant created his work by "iterative or exact duplication of substantial portions of the copyrighted work." *Johnson,* 623 F. Supp. at 1493. The plaintiff can establish that the defendant used the copyrighted work in preparing the alleged copy by showing that the defendant had access to the copyrighted work and that the works are similar enough to infer use of the copyrighted work. *Id.* The factfinder's focus under the "iterative" test of substantial similarity shifts from the ordinary observer analysis to an analysis of the "quantitative and qualitative evidence of similarities" in the copyrighted and allegedly infringing computer programs. *Id.; see* Whelan Assoc., Inc. v. Jaslow, 797 F.2d 1222, 1236 (3d Cir. 1986) (adopting single substantial similarity test in which both lay and expert testimony is available).

[46] NIMMER, THE LAW OF COMPUTER TECHNOLOGY 1-51 (1985). Computer programs have several characteristics that make the distinction between expression and unprotected ideas difficult to determine in cases of nonliteral copying. *Id.* First, limited and highly structured

puter programs both describe and implement a process, the expression and the underlying process or idea in a computer program are related more closely than in traditional forms of expression.[47] While copyright law does not preclude protection of a program merely because the program's expression implements a process, the expression in computer programs is different from the expression in most other literary works.[48] In contrast to ordinary literary works, the user of a program does not perceive the program's expression directly but perceives only a product of the program's expression.[49] This derivative nature of a program's expression and the expression's close association with the process or idea of the program, therefore, create conceptual difficulties when applying the idea-expression distinction to computer programs.[50]

To complicate the already daunting task of determining the proper scope of copyright protection for computer programs, courts often have trouble understanding the nature of computer programs.[51] Numerous programming languages,[52] different kinds of pro-

programming languages restrict the forms of expression available to programmers. *Id.* Second, the functional or utilitarian purposes of a program also may restrict the number of expressive options available to programmers. *Id.* Third, programmers' common goal of efficient programming encourages similarity in coding. *Id.* The scope of copyright protection should reflect the kind of program involved. *Id.* Some programs involve complex artistic outputs and have a broad range of creative expression, while other programs that direct the internal workings of computers, are more standardized and have a more restricted range of expression. *Id.* at 1-52. The scope of copyright protection should reflect, therefore, the extent of unique expression available to create a particular program. *Id.* at 1-51, -52.

[47] *See id.* at 1-10. The relationship between process and expression is closer in computer software than possibly any other creative industry. *Id.* The connection between idea and expression in computer software, therefore, is not comparable to books or articles that describe a process. *Id.*

[48] *See* M. Kramer Mfg. Co. v. Andrews, 783 F.2d 421, 435 (4th Cir. 1986) (copyright law does not preclude protection of computer programs merely because words of program implement process or system); Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521, 525 (9th Cir. 1984) (noting that program's utilization of its expression while implementing process does not affect program's copyrightability); CONTU FINAL REPORT, *supra* note 4, at 21 (same).

[49] *See* 1 Davis, *Computer Software—The Final Frontier: Clones, Compatibility and Copyright,* 2 THE COMPUTER LAWYER No. 6, 1, 7 (1985) (describing unique form of expression in computer software).

[50] *See supra* notes 46-49 and accompanying text (discussing difficulties in applying idea-expression dichotomy doctrine in computer software context).

[51] *See* Q-Co Indus., Inc. v. Hoffman, 625 F. Supp. 608, 620 (S.D.N.Y. 1985) (expressing trouble courts have in understanding "esoterica of bytes and modules" in computer software); SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985) (acknowledging courts' difficulty in determining which lines of program constitute misappropriation).

[52] *See* Ralston & Meek, ENCYCLOPEDIA OF COMPUTER SCIENCE 1173 (1976) (by 1973 programmers were using more than 200 high level languages); *infra* notes 56-59 and accompanying text (defining high-level programming languages). Examples of programming languages that have had significant importance include APT (Automatically Programmed Tools), BASIC (Beginners All-purpose Symbolic Instruction Code) FORTRAN (FORmula TRANslation), IPL-V (Information Processing Language V), COBOL (Common Business-Oriented Language), Algol (Algorithmic Language), and Lisp (List Processing). Ralston & Meek, *supra,* at 1173. Major types of programming languages include: (1) Assembly, or symbolic, machine languages, (2) Macroassembly languages, (3) Problem-oriented languages, (4) Procedure-oriented languages,

grams,[53] and various mediums for the storage of programs[54] make understanding the computer environment abstruse for the inexperienced. Programming languages are classified into several levels corresponding to the language's dependence on a particular machine's characteristics.[55] The three basic categories of programming languages are high-level languages, low-level languages, and machine language.[56] Computers operate by executing binary instructions, which are instructions composed of a sequence of zeroes and ones.[57] Binary instructions, known as "machine language" or "object code," are extremely difficult to understand

(a) Algebraic languages (numerical computation), (b) String-manipulating languages (text manipulation), (c) Simulation languages, and (d) Multipurpose languages. SIPPL, MICROCOMPUTER DICTIONARY 361 (2d ed. 1981).

[53] *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983), *cert. dismissed,* 464 U.S. 1033 (1984). Computer programs are categorized generally by function as application or operating system programs. *Id.* Application programs perform a specific function for the user, like word processing. *Id.* Operating system programs control the internal functions of the computer or control the use of application programs. *Id.*

[54] *See* Ralston & Meek, *supra* note 52, at 883-916. Computer systems have different levels of storage, known as memory. *Id.* The basic categories of memory are main memory (MM) and auxiliary memory (AM). *Id.* at 900. The major difference between MM and AM is that a computer only can execute instructions that are in MM. *Id.* AM comprises all other memory, and a computer must fetch instructions from the AM into the MM before the computer can execute them. *Id.* Different forms of AM include magnetic tapes, cassette tapes, drums, fixed-head disks, moving-disks, data cells, photocopy storages, and large core storages. *Id.* One common form of AM is the "floppy disk," which is a flexible magnetic disk that resembles a phonograph. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d at 1243 (describing floppy disks). MM is always read/write memory. *Id.* The term "read" refers to a computer's ability to copy or accept data from memory registers, AM, and input devices like keyboards, and barcode scanners. *See* SIPPL, *supra* note 52, at 379 (defining term "read"). The term "write" means to record data permanently or transiently in a storage device. *Id.* Read-only memory (ROM) is memory from which a computer can read instructions but to which the computer cannot record or "write" instructions. *See id.* (defining read-only memory). ROM usually refers to instructions or data stored on a semi-conductor chip incorporated into the circuitry of the computer. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983) (ROM is internal permanent memory device consisting of semi-conductor chip that is incorporated into circuitry of computer), *cert. dismissed,* 464 U.S. 1033 (1984). EPROMS (Erasable Programmable Read Only Memory) are ROMs that are erasable and reprogrammable. *Id.*

[55] Seidman & Flores, THE HANDBOOK OF COMPUTERS AND COMPUTING 331 (1984). The "level of discourse" of a programming language is the language's distance from the underlying properties of the machine that implements the program. *Id.*

[56] *See* BORKING, *supra* note 2, at 408-13 (1985) (discussing different levels of programming languages); *see also* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983) (discussing three levels of computer language available for writing computer programs), *cert. dismissed,* 464 U.S. 1033 (1984).

[57] *See* BORKING, *supra* note 2, at 17-18 (central processing unit (CPU) processes binary instructions); *see also* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983) (central processing unit (CPU) of computer can only implement binary instructions (object code)), *cert. dismissed,* 464 U.S. 1033 (1984). The Third Circuit in *Franklin* noted that the lowest level of computer language is machine language, known as object code, which only uses two symbols, 0 and 1. *Id.* The *Franklin* court used the example that the instruction "01101001" causes the Apple II computer to add two numbers and save the result. *Id.*

and cumbersome to use.[58] For this reason, programmers usually write programs in either low or high-level languages, known as "source code."[59] When a programmer writes a program in source code, the programmer must run another program, called a compiler or an interpreter, to translate the source code into binary instructions so the computer can execute the program.[60]

Writing the source code is only one step in the development of a commercial software program.[61] The process of creating a commercially marketable program usually involves a series of stages, with each stage comprising increasingly detailed steps toward achieving the intended result.[62] Each stage in the development process can include original works of authorship.[63] Although each programmer has an individual style for developing a program, most programmers develop programs in a similar manner. The first step involves defining the end result and program objectives.[64] The programmer usually draws a flow chart or creates an outline that describes the different functions or components of the program[65] and may describe the modules or subroutines necessary to accomplish the desired result.[66] At

[58] Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983), *cert. dismissed,* 464 U.S. 1033 (1984).

[59] *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1243 (3d Cir. 1983) (programmers usually write programs in source code, which is more comprehensible to humans), *cert. dismissed,* 464 U.S. 1033 (1984). Low level languages consist of alphanumeric labels like "ADC," which instructs an Apple II computer to "add with carry." *See* SIPPL, *supra* note 52, at 361 (low-level language is programming language whose instructions bear one-to-one relationship with machine code). In high-level language, the instructions are familiar notations, approach ordinary English or conventional mathematical notations, and usually correspond to several object code instructions. *See id.* (high level languages consist of familiar notations); *Franklin,* 714 F.2d at 1243 (high level languages like BASIC and FORTRAN use English words and symbols). An advantage of high-level languages is that they are similar on different computers. SIPPL, *supra* note 52, at 74. Low-level programming languages, however, may vary substantially with each type of computer. *Id.*

[60] *See* SIPPL, *supra* note 52, at 75, 196 (defining compiler and interpreters). Once the programmer has written a program in source code, the programmer can run a compiler program that will convert the program to a sequence of machine instructions that become the binary object code. *Id.* An interpreter performs a function similar to a compiler, but an interpreter will translate the instructions one at a time and execute them immediately. *Id.* at 196.

[61] *See* FRANK, *supra* note 10, at 21-22 (1982) (discussing different stages in development of computer program).

[62] *Id.; see* SAS Inst., Inc. v. S & H Computer Sys., 605 F. Supp. 816, 828 (M.D. Tenn. 1985) (author of program begins with broad and general statement of purpose for program and breaks each task into successively more detailed tasks until at lowest levels programmer translates instructions into source code).

[63] Brief of Amicus Curiae at 10, Whelan Assoc., Inc. v. Jaslow Dental Laboratories, 797 F.2d 1222 (3d Cir. 1986). The significance of the multi-stage development process of computer programs is that each stage involves authorship in the programmer's selection of the program's structure, sequence, pattern, and organization. *Id.*

[64] *See* LEATHRUM, FOUNDATIONS OF SOFTWARE DESIGN 8 (1983) (first step in lifecycle of computer software is defining purpose and intended results of software).

[65] SIMPSON, DESIGN OF USER-FRIENDLY PROGRAMS FOR SMALL COMPUTERS 18-25 (1985).

[66] *See* Ralston & Meek, ENCYCLOPEDIA OF COMPUTER SCIENCE, *supra* note 52, at 1372.

some point, the programmer must define the user interface, which controls the manner in which the computer interacts with the user including the visual display of the input and output formats that appear on the screen, and the means by which the program enables the user to control the computer.[67] The programmer also will establish a set of detailed statements that define the necessary instructions for the computer to execute.[68] Finally, the programmer will translate the detailed steps or program specifications into the source code using one of many possible programming languages.[69] In addition to translating the program specifications into code, a programmer must continually "debug" the program, which consists of locating and correcting errors in the computer program.[70] Coding is the final step in the creation of a computer program and represents approximately one fifth of the total development cost.[71] Thus, at each stage in the development of a

A subroutine is part of a program that is logically separate and performs a specific task necessary for the program's execution. *Id.* A subroutine ordinarily represents the implementation of a process that the program utilizes many times. *Id.*

A program module is "a logically self-contained and discrete part of a larger program." *Id.* at 943. A correctly designed module accepts well-defined input, executes a specific set of instructions, and produces an output well-defined as to content and structure. *Id.* Modular programming allows a programmer to break complex tasks into smaller and less complex subtasks, facilitating the design and testing of programs. *Id.* Modular programming greatly enhances the popular technique of "top-down" program design. *Id.* Top-down program design begins with the most general definition of the program's function and proceeds with a sequence of increasingly detailed specifications. *Id.*

[67] *See* SIMPSON, *supra* note 65, at 70 (1985) (discussing user interface as "human-computer interface"); *see also* Russo & Derwin, *Copyright in the "Look and Feel" of Computer Software*, 2 THE COMPUTER LAWYER No. 2, 1, 1 (1985) (design and presentation of program, also known as "look and feel" of computer software, are becoming increasingly important for computer software). The presentation of a program refers to the visual screens and the manner in which the program presents useful information to the user. *Id.* The user-interface, often referred to as the "look and feel" of a program, can be the most important aspect in the commercial success of a program. *See* HECKEL, THE ELEMENTS OF FRIENDLY SOFTWARE DESIGN 25 (1984) (quality of user interface is important for success of software project); SIMPSON, *supra* note 66, at 62-63 (researchers point out that programs have failed because of improper design for intended users). The display design issues are significant factors for a program's success in the user community. VICK & RAMAOORTHY, HANDBOOK OF SOFTWARE ENGINEERING 538-39 (1984) (noting that human factors are integral part of display design) (quoting LUXENBURG & KUEHN, DISPLAY SYSTEMS ENGINEERING (1968)). The human factors that programmers must consider for designing the program's display include the user's perception, comprehension, viewing environment, and psychological factors. *Id.*

[68] CASSEL, THE STRUCTURED ALTERNATIVE: PROGRAM DESIGN, STYLE, AND DEBUGGING 81 (1983).

[69] *Id.; see supra* note 52 (listing several source code programming languages).

[70] *See* SIPPL, *supra* note 52, at 106 (defining debugging as process of isolating and removing errors or malfunctions from computer or computer program).

[71] FRANK, *supra* note 10, at 22. *See* Ralston & Meek, *supra* note 52, at 1159 (coding generally refers to writing and debugging programs for given program specifications, while programming includes task of preparing program specification in addition to writing program). The term "coder" sometimes has pejorative connotations to describe a person who engages exclusively in writing programs based on the program specifications of others. *Id.*

computer program, the programmer refines his idea into an increasingly detailed expression of the idea, and courts must decide at what point the programmer has transformed the idea of the program into copyrightable expression.[72]

In determining what constitutes protectible expression in computer software, judicial decisions applying the '76 Act to computer programs have established that copyright law protects both the source and object code of computer programs,[73] regardless of whether the programmer has stored the code on electromagnetic disk, tape, or in Read Only Memory (ROM).[74] In addition, courts have established that copyright protects computer programs regardless of whether the program performs a specific function for the user, such as word processing, or controls the internal operation of the computer.[75] While copyright law clearly protects the literal source and object code of a computer program, the extent to which the copyright of a computer program protects the nonliteral elements of a program like the structure, organization, sequence, and user interface remains unresolved.[76]

[72] *See supra* notes 35-45 and accompanying text (discussing fundamental principle of copyright law that copyright protects expression of idea only).

[73] *See* Apple Computer, Inc. v. Formula Int'l Inc., 725 F.2d 521 (9th Cir. 1984) (finding copyright protection for object code of computer program); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983) (copyright protects computer programs whether in object or source code), *cert. dismissed,* 464 U.S. 1033 (1984); GCA Corp. v. Chance, 217 U.S.P.Q. 718, 720 (N.D. Cal. 1982) (copyright law treats source and object code as one work, so copyright in source code protects object code); CONTU FINAL REPORT, *supra* note 4, at 21 (machine language version of program is copy of source code because programmer can produce copy of source code from machine language version of program). In *Apple Computer, Inc. v. Franklin Computer Corp.,* the Ninth Circuit held that copyright protects a computer program regardless of whether the program is in source or object code. *Franklin,* 714 F.2d at 1249. The *Franklin* court found that CONTU intended copyright to protect both the instructions communicated to people and those communicated to machines. *Id.* at 1248.

[74] *See, e.g.,* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (9th Cir. 1983) (copyright of computer program protects object code stored in ROM), *cert. dismissed,* 464 U.S. 1033 (1984); Williams Elec., Inc. v. Arctic Int'l, Inc., 586 F.2d 870, 874 (3d Cir. 1982) (same); Midway Mfg. Co. v. Strohon, 453 F. Supp. 741, 749-53 (N.D. Ill. 1983) (copyright of computer program protects program whether stored on disk, tape, or ROM). The Ninth Circuit in *Apple Computer, Inc. v. Franklin Computer Corp.* found that programs stored in ROM were "fixed" within the meaning of the 1976 Act and were within the protection of copyright law. *Franklin,* 714 F.2d at 1250. *See supra* note 54 (discussing different kinds of memory storage devices).

[75] *See* Apple Computer Inc. v. Formula Int'l, Inc., 725 F.2d 521, 525 (9th Cir. 1984) (finding copyright protection for operating systems); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983) (copyright law does not *per se* exclude protection of operating system programs), *cert. dismissed,* 464 U.S. 1033 (1984). In *Apple Computer, Inc. v. Franklin Computer Corp.,* the *Franklin* court did not agree with Franklin's argument that applications software is protectible while systems software is not. *Franklin,* 714 F.2d at 1249. The *Franklin* court reasoned that the utilitarian aspect of a work will not preclude the work from copyright protection. *Id.* at 1250; *see supra* note 53 (discussing application programs and operating systems).

[76] *See* Whelan Assoc., Inc. v. Jaslow Laboratory, Inc., 797 F.2d 1222 n.1 (3rd Cir. 1986) (using terms "structure," "sequence," and "organization" interchangeably).

Although the case law is sparse, a few courts have addressed the issue of which nonliteral elements of a computer program constitute protectible expression.[77] In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*,[78] for example, the United States District Court for the Eastern District of Pennsylvania considered which elements of a copyrighted program constitute protectible expression.[79] The defendant in *Whelan*, Jaslow Dental Laboratory, Inc. (Jaslow), hired Strohl Systems to design a software system for various business operations of Jaslow's dental laboratory.[80] Jaslow explained to Elaine Whelan, a fifty-percent shareholder and employee of Strohl Systems, the features, capabilities, and functions that would be useful in operating a dental laboratory, and Elaine Whelan designed a computer program, the Dentalab System, for operating the dental laboratory.[81] Elaine Whelan wrote the program in Event Driven Language (EDL) for an IBM/ Series 1 computer.[82] Both parties agreed that Strohl Systems could adapt,

[77] *See, e.g.*, Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1248 (holding that copyright protection for computer programs extends to their structure, sequence, and organization); Q-Co Indus., Inc. v. Hoffman, 625 F. Supp. 608, 616 (copyright infringement case addressing whether structure and organization of computer program constituted protectible expression); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1497 (D. Minn. 1985) (addressing whether nonliteral conversion of programs from one microprocesser language to another constitutes copyright infringement); Williams v. Arndt, 626 F. Supp. 571, 577-78 (D. Mass. 1985) (addressing whether translation from English to source code constitutes copyright infringement); SAS Inst., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985) (same); *infra* notes 77-169 and accompanying text (discussing cases that have addressed issue of protectible expression in computer software).

[78] 609 F. Supp. 1307 (E.D. Pa. 1983), *aff'd*, 797 F.2d 1222 (3d Cir. 1986).

[79] *Id.* at 1320. In *Whelan*, the United States District Court for the Eastern District of Pennsylvania addressed the question whether copyright protection in plaintiff's software extended beyond a program's source and object code. *Id.* The source and object codes of the plaintiff's and defendant's programs were not similar because the defendant wrote his program in a different programming language for a different computer. *Id.* The *Whelan* court, therefore, had to consider whether nonliteral elements of the plaintiff's program were protectible. *Id.*

[80] *Id.* at 1317. In *Whelan*, Jaslow attempted to develop a program to operate his dental laboratory, Jaslow Dental Laboratory, Inc. (Jaslow Laboratory). *Id.* at 1309. After failing to develop a usable program, Jaslow hired Strohl Systems to design a computer program for the Jaslow Laboratory. *Id.* Strohl Systems specialized in designing custom software programs. *Id.* Under the initial agreement between Strohl Systems and Jaslow Laboratory, Strohl Systems was to receive $18,000 for designing a custom dental laboratory software package. *Id.* The parties agreed that Strohl Systems was to retain ownership of all software developed for Jaslow. *Id.* at 1310. Strohl Systems designated Elaine Whelan as the person in charge of designing and developing the program. *Id.*

[81] *Id.* at 1310. To design the Dentalab System in *Whelan*, Elaine Whelan studied extensively the business operations of the Jaslow Laboratory. *Id.* Whelan studied in detail the methods that Jaslow Laboratory had used to receive, process and deliver orders. *Id.* Elaine Whelan also studied other business operations of the Jaslow Laboratory like invoicing, billing, accounting and inventory control. *Id.* Additionally, Elaine Whelan conducted substantial independent research into the methods of operation employed by other dental laboratories. *Id.*

[82] *Id.* Event Driven Language (EDL) is a recognized computer programming language. *Id.*

market, and sell the program to other dental laboratories.[83] Subsequently, Elaine Whelan acquired all of Strohl System's interest in the software, and later transferred her interest to the plaintiff, Whelan Associates, Inc. (Whelan Associates), a software development corporation of which Elaine Whelan was the president and the controlling shareholder.[84] Whelan Associates and Jaslow entered into a marketing agreement in which Jaslow would market and sell the Dentalab System and would receive a commission for all of its sales.[85] The parties agreed that upon the termination of the marketing agreement, Jaslow would return all copies of the Dentalab System's source and object code to Whelan Associates, and that all rights to market the program would revert to Whelan Associates.[86]

Although under the marketing agreement, Whelan Associates was to have exclusive possession of all copies of the Dentalab System's source code, Jaslow obtained a copy of the Dentalab System source code for the IBM/Series 1 without Whelan Associates' consent.[87] Jaslow formed a corporation, Dentcom, to market computer programs, including the Whelan Associates' Dentalab system, to the dental laboratory industry.[88] Utilizing the copy of Whelan Associates' source code and the assistance of an expert programmer, Jaslow developed a dental laboratory program in the BASIC programming

[83] *Id.* at 1310. In *Whelan,* both parties agreed that Strohl Systems would design a program that a skilled programmer easily could adapt for use by other dental laboratories in the industry so that Strohl Systems could market the dental laboratory software to other dental laboratories, and Jaslow would receive 10% of the sales price of packages sold to other dental laboratories. *Id.* The agreement provided, however, that Strohl Systems would receive all the proceeds for any modifications to the system, which Strohl Systems would sell separately from the basic package. *Id.* Both Strohl Systems and Jaslow understood that the operations of dental laboratories were substantially similar throughout the industry and that any system developed for Jaslow Laboratories would readily be adaptable to many other dental laboratories. *Id.*

[84] *Id.* at 1317. In *Whelan,* Elaine Whelan formed a corporation, Whelan Associates, Inc. (Whelan Associates), in November 1979 as a corporation to market dental laboratory software under the tradename Dentalab. *Id.*

[85] *Id.* at 1312-13. In *Whelan,* Strohl Systems and Jaslow Laboratories originally entered into an agreement which provided that ownership of the software program developed by Strohl Systems would remain in Strohl Systems and that Strohl Systems would market the program to other dental laboratories. *Id.* Jaslow Laboratories agreed to demonstrate the program to prospective purchasers and in return would receive a 10% commission on systems sold to other laboratories. *Id.* at 1310. In 1979 Strohl Systems began marketing the dental laboratory software package under the trade name Dentalab. *Id.* at 1311. In 1980, after Whelan Associates had acquired marketing rights to the program, Whelan Associates and Jaslow Laboratory entered into an agreement whereby Jaslow would receive 35% of the gross price of any Dentalab programs sold and 5% for any modifications. *Id.* at 1317-18. Although the agreement between Whelan Associates and Jaslow provided that Jaslow owned its own specialized IBM/Series 1 program, the district court in *Whelan* concluded that Jaslow Laboratory did not own any proprietary interest in the source or object code designed by Elaine Whelan for use in other dental laboratories. *Id.* at 1318.

[86] *Id.*

[87] *Id.* at 1314.

[88] *Id.* at 1315.

language for an IBM personal computer (IBM PC).[89] Jaslow, through Dentcom, subsequently marketed and sold his dental laboratory program that closely resembled the program designed by Elaine Whelan.[90] Whelan Associates sued Jaslow for copyright infringement of Whelan Associates' IBM/Series 1 program.[91]

In determining whether Jaslow's program infringed Whelan Associates' copyright in the Dentalab System program, the *Whelan* court noted that if copyright protection extends only to the source and object codes of a program, then Jaslow had not infringed Whelan Associates' copyright because Elaine Whelan and Jaslow had created the two programs in different programming languages on different computers, and the source and object codes of the two programs were, therefore, different.[92] When discussing the distinction between the idea and expression in computer programs, the district court in *Whelan* noted that the idea of a computerized system for operating

[89] *Id.* at 1314-15. In *Whelan*, around May or June of 1982, Jaslow unsuccessfully attempted to write a program in the BASIC programming language for an IBM personal computer (IBM PC). *Id.* The IBM PC was less expensive and smaller than the IBM/System 1 computer, and, therefore, programs designed for the IBM PC had the greatest potential for commercial success. *Id.* In attempting to write the program, Jaslow used the Dentalab source code and tried to duplicate the identical functions, format of the visual screens, methods of collating, file structure, and abbreviations contained in the Dentalab program for the IBM/Series 1 computer. *Id.* Because Jaslow was unable to design and write a usable program himself, in August 1983 Jaslow hired a skilled programmer who succeeded in developing a program for the IBM PC. *Id.* Whelan Associates also designed a version of the Dentalab program for the IBM PC that performed most of the same functions as the Dentalab program designed for the IBM/Series 1 computer. *Id.*

[90] *Id.* at 1315-16. The district court in *Whelan* found that Jaslow Laboratory and Dentcom marketed Jaslow's version of the dental laboratory program under the names Dentalab and Dentlab. *Id.* at 1315. The district court found that Jaslow Laboratory and Dentcom used the name "Dentlab" to mislead the dental laboratory industry and prospective purchasers into thinking that the Dentcom system was a version of Whelan Associate's Dentalab program. *Id.* The district court in *Whelan* court also found that Dentcom's use of the name "Dentlab" had confused some potential customers. *Id.*

[91] *Id.* at 1320. In *Whelan,* Whelan Associates alleged that Jaslow had infringed Whelan Associates' copyright in the Dentalab program by selling copies of the Dentalab system after the termination of the parties' marketing agreement, which provided that all marketing rights reverted to Whelan Associates. *Id.* Whelan Associates also alleged that Jaslow had infringed Whelan Associates' copyright in the Dentalab program by developing and marketing a dental laboratory program for the IBM PC that was very similar to the Dentalab program. *Id.*

[92] *Id.* at 1320. In *Whelan,* the district court found that Jaslow had not translated the source code from the IBM/System 1 program verbatim. *Id.* at 1315. The district court noted, however, that Jaslow could copy the program without literally translating the source code from EDL into BASIC. *Id.* Noting that literal translation of the source code was an inefficient way to copy the Dentalab program, the district court suggested that to copy a program from one programming language into a different programming language, one would study the manner in which data flows sequentially from one function to another. *Id.* at 1321. A plagiarist would then copy the program's particular manner of operation by using commands in the second programming language that correspond functionally to the commands used in the original program. *Id.* at 1321.

a dental laboratory is not subject to copyright protection.[93] The expression of an idea in a computer program, according to the district court in *Whelan*, is the manner in which the program regulates, controls, and operates the computer in collecting and manipulating data to produce useful information.[94] Consequently, the district court in *Whelan* implicitly recognized that the expression in a computer program includes the structure and organization of the code.[95]

Affirming the district court's opinion in *Whelan*, the United States Court of Appeals for the Third Circuit explicitly held that copyright protection can extend beyond a computer program's literal code to the program's structure, organization, and sequence.[96] The Third Circuit formulated the following distinction between expression and idea in utilitarian works: "the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that function or purpose would be part of the expression of the idea."[97] Because other programs existed that contained different structures while performing the same function as the Dentalab and Dentcom programs, the Third Circuit concluded that the Dentcom's detailed structure was part of the program's protectible expression.[98]

[93] *Id.*

[94] *Id.* The district court in *Whelan* formulated the following definition of the expression of the idea in a computer program:

> The 'expression of the idea' in a computer program is the manner in which the program operates, controls and regulates the computer in receiving, assembling, calculating, retaining, correlating, and producing useful information either on a screen, print-out or by audio communication.

Id.

[95] *See id.* The district court in *Whelan* arguably was referring implicitly to the program's structure and organization because the structure and organization of a program dictate the method and manner in which the program will operate and manipulate data. *Id.*; *cf.* SAS Inst. Inc. v. S & H Computer Sys., 605 F. Supp. 816, 826 (M.D. Tenn. 1985) (noting that copyright protection of structure and organization of computer program is important because appropriation of proven design and structure of program gives competitors advantage of avoiding false starts, redesigns and other fruitless efforts). The district court in *Whelan* noted that copyright law protects the expression of an idea even though one must alter or refine the expression to adapt a program for use on other computers that respond to different instructions. *Whelan*, 609 F. Supp. at 1320.

[96] Whelan Assoc., Inc. v. Jaslow, 797 F.2d 1222, 1248 (3d Cir. 1986).

[97] *Id.* at 1236. In concluding that copyright protection for computer programs can extend to the structure, sequence, and organization of a program, the Third Circuit in *Whelan* disagreed with the United States District Court for the Northern District of Texas in *Synercom Technology, Inc. v. University Computing Co.* to "the extent that *Synercom* rested on the premise that there was a difference between the copyrightability of sequence and form in the computer context and in any other context." *Id.* at 1240. *See* Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003, (N.D. Tex. 1978) (holding that input formats of computer program were not subject to copyright protection because structure and sequence of input formats were indistinguishable from underlying ideas); *Whelan*, 797 F.2d at 1240 (noting that Congress intended ordering and sequencing to be protectible in appropriate circumstances).

[98] *Id.* at 1238-39.

In addition to the Third Circuit's opinion in *Whelan* extending copyright protection to the structure and organization of a computer program, the United States District Court for the Middle District of Tennessee in *SAS Institute, Inc. v. S & H Computer Systems, Inc.*[99] also has considered whether copyright protects these elements of a program.[100] The plaintiff in *SAS,* SAS Institute, Inc. (SAS), was a corporation that marketed a computer program for performing statistical analysis on IBM computers.[101] SAS owned a valid copyright for its statistical program marketed under the trademark "SAS."[102] SAS had begun testing a version of the SAS program for the Digital Equipment Corporation's (DEC) VAX computer.[103] S & H Computer Systems, Inc. (S & H) decided to develop a program like the SAS program to run on the VAX computers.[104] S & H licensed a copy of SAS' program to obtain detailed technical information, including the source code, to use in preparing S & H's conversion of the SAS program to run on the VAX computers.[105]

[99] 605 F. Supp. 816 (M.D. Tenn. 1985).

[100] SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985).

[101] *Id.* at 818, 819. In *SAS,* the copyright infringement involved SAS' computer program version 79.5, which took more than five years and 18 man-years of labor to develop. *Id.* at 818. The SAS program Release 79.5 operated on IBM and IBM-compatible computers manufactured by IBM's competitors. *Id.*

[102] *Id.*

[103] *Id.* at 818. Responding to user interests, SAS began developing non-IBM versions of the SAS program in 1981. *Id.* By the time of the trial, SAS had sent over 200 copies of a VAX version of the SAS program to test sites. *Id.* The defendant, S & H Computer Systems, Inc. (S & H) had attempted to design and develop its own statistical program—PASQUEL. *Id.* After determining that the program would be much more difficult to market than a program modeled after the SAS program, S & H abandoned the project. *Id.* at 819.

[104] *Id.* at 819. In *SAS,* several professors at Vanderbilt University wanted to use the SAS program. *Id.* Since Vanderbilt did not have an IBM computer and SAS had not yet announced its non-IBM version of the SAS program, the professors formed a limited partnership under the name "Portable Statistical Analysis Systems, Ltd." (PSAS, Ltd.) to develop a conversion of the SAS program to operate on the VAX computer. *Id.* at 820. In an agreement between PSAS, Ltd. and S & H, the parties agreed to develop a computer software package similar to SAS. *Id.*

[105] *Id.* at 820-21. In *SAS,* SAS did not sell copies of its statistical analysis system, but sold licenses to use copies of the program. *Id.* When S & H licensed the SAS program, S & H did not disclose its purpose in acquiring a license for the SAS program. *Id.* The *SAS* court found that S & H breached an implied duty of good faith and fair dealing by licensing the SAS program for the sole purpose of obtaining proprietary materials, which S & H used to develop a competing program. *Id.* at 827-28. SAS and S & H entered into a licensing agreement whereby S & H agreed to use the SAS program on a specific IBM machine located at Tennessee State University in Nashville. *Id.* Under the terms of the license agreement, S & H agreed not to further distribute the SAS program, allow time sharing use of the SAS program, or make any copies of the program except for back-up purposes. *Id.* The *SAS* court found that S & H had violated the licensing agreement by licensing the program with the intent to use technical information about the SAS program to develop a similar program for the VAX computer. *Id.*

SAS sued S & H for copyright infringement.[106]

At the conclusion of the trial, the *SAS* court found that S & H had infringed SAS' copyright.[107] The *SAS* court held that S & H's INDAS program, S & H's version of the SAS program, was an unlawful copy of the SAS program that violated SAS' valid copyright in the program.[108] The *SAS* court rejected S & H's argument that S & H merely had copied unprotectible ideas by copying only the organization and structure of the SAS program.[109]

To support its holding that the INDAS program was an illegal copy of the SAS program, the *SAS* court articulated numerous indicia of direct or substantially similar copying of the SAS program.[110] The *SAS* court considered as evidence of copying S & H's use of a close paraphrase of the SAS program's organization, and the "slavish copying" by S & H of the SAS

[106] *Id.* at 828. In *SAS,* SAS alleged that S & H had made an unknown number of unauthorized copies of the SAS source code during the development of its conversion program, INDAS. *Id.* Furthermore, SAS claimed that the INDAS program was either a "copy" or a "derivative work" based on the SAS program, and that in either case, S & H had infringed SAS' copyright. *Id.*

[107] *Id.* at 821. In *SAS,* since the IBM computer on which S & H had agreed exclusively to operate the SAS program could not execute source code instructions in ASCII, the *SAS* court held that a copy of source code, which S & H had translated into ASCII, could not constitute a back-up copy of the SAS program. *Id.* Consequently, the *SAS* court found that S & H had breached the licensing agreement by making an unauthorized copy. *Id.; see supra* note 105 and accompanying text (discussing terms of licensing agreement between SAS and S & H).

[108] *SAS,* 605 F. Supp. at 830. In addition to 44 specific instances of copying proven at trial, the court in *SAS* also found that copying of the SAS program's organization and structural detail pervaded the S & H program. *Id.* The court noted that SAS' expert, Dr. Peterson, testified that in his opinion, the early source code for the S & H program was substantially similar to the SAS source code. *Id.* at 822. The court, therefore, found that S & H's program was substantially similar to the SAS program and infringed the copyright in the SAS program. *Id.* at 830.

[109] *Id.* at 829. In *SAS,* S & H argued that by writing independently the code for each of the lowest level tasks, S & H had copied only the organizational scheme of the SAS program, which S & H asserted was an unprotectible idea of the program. *Id.* The *SAS* court did not believe S & H's assertion that S & H had created its source code independently. *Id.* The *SAS* court noted that even if S & H had distributed an outline of the SAS program to programmers, and S & H programmers independently created the INDAS source code based on the outline, such a process would have duplicated the expression in the SAS program. *Id.; see infra* note 118 and accompanying text (discussing case in which employees wrote "original" text for textbook based on outlines and *Meredith Corp. v. Harper & Row Publishers, Inc.* court concluded that such process resulted in duplication of expression). The *SAS* court noted that whether similarities in a program are similarities of ideas or similarities of expression is a question of fact. *SAS,* 605 F. Supp. at 829. The *SAS* court found that as a matter of fact S & H had duplicated the expression of the SAS program. *Id.*

[110] *SAS,* 605 F. Supp. at 822. In *SAS,* the district court relied on SAS' expert witness who testified that the S & H program contained at least 44 specific instances of direct copying from the SAS source code and another 18 examples of source code "similar" to the SAS code. *Id.* The *SAS* court rejected S & H's assertion that 44 instances of literal copying were insubstantial as a matter of law. *Id.* at 830.

program's structural detail.[111] The *SAS* court noted that S & H's utilization of the SAS program's source code in preparing the INDAS program was an additional indication of copying.[112] In addition, the *SAS* court regarded the existence of undocumented options from the SAS program in the INDAS program as further evidence of copying.[113] S & H's lack of design documentation led the *SAS* court to conclude that S & H had adopted SAS' design documentation in detail and had appropriated a large part of the SAS program's design.[114] Finally, the *SAS* court noted that evidence of S & H's attempt to disguise its copying of the SAS program and S & H's copying of nonfunctional features in the SAS program contributed to the *SAS* court's holding of infringement.[115]

Addressing the issue of copying a program's structure and organization, the *SAS* court found that S & H had saved substantial time and effort, avoiding the risk and expense of "dead ends, redesigns, and other fruitless efforts" by copying the proven structure of the SAS program.[116] The *SAS* court cited the reasoning of the United States District Court for the Southern District of New York in *Meredith Corp. v. Harper & Row Publishers, Inc.*[117] The *Meredith* court found that the process in which Meredith employees distributed portions of an outline of a textbook to freelance writers who wrote "original" text based on the outline resulted in the copying of protectible expression and not ideas.[118] The court in *SAS*, therefore, concluded that under the *Meredith* court's reasoning, even if S & H had not copied from the source code of the SAS program, S & H's use of a detailed outline of the SAS pro-

[111] *Id.* at 826.

[112] *Id.* at 826. In *SAS*, the district court found that S & H's programmers systematically and pervasively relied on the SAS source code when developing the S & H program. *Id.*

[113] *Id.* at 824. The district court in *SAS* noted that SAS had not documented many elements of the SAS user language such as the set of commands and instructions that enable the user to make the program achieve certain results. *Id.* According to the *SAS* court, S & H could only have learned of the existence of these undocumented options by close examination of SAS' source code. *Id.*

[114] *Id.* at 824. In *SAS*, S & H produced a document known as "DOCOUT," which S & H claimed was the design document for the INDAS program. *Id.* The *SAS* court, however, found that this document was an extraction of the SAS source code's "comments," which describe the function of the accompanying source code. *Id.* The *SAS* court particularly noted that the DOCOUT document dealt almost entirely with "interface routines" and that the S & H program contained 69 out of the 70 "interface routines" in the SAS program. *Id.* Consequently, the *SAS* court found that S & H had appropriated the design of the SAS source code before and during the creation of DOCOUT. *Id.*

[115] *Id.* at 823-24.

[116] *Id.* at 826.

[117] 378 F. Supp. 686 (S.D.N.Y.), *aff'd*, 500 F.2d 1221 (2d Cir. 1974), *permanent injunction entered*, 413 F. Supp. 385 (S.D.N.Y. 1975).

[118] *Id.* at 386-87.

gram in preparing S & H's program would have constituted copyright infringement.[119]

In a case similar to *Whelan* and *SAS,* the United States District Court for the District of Minnesota in *E.F. Johnson Co. v. Uniden Corp. of America*[120] recognized that copyright protection extends beyond a computer program's literal code.[121] The plaintiff in *Johnson,* E.F. Johnson Co. (Johnson), a manufacturer of land-based communication systems, developed a logic trunked radio system (LTR), which established an efficient means of two-way communication between mobile radio units and base stations.[122] The principal component of Johnson's LTR system was the computer software stored in the mobile radios and base stations.[123] Johnson owned a valid copyright for the software contained in the mobile units.[124] The defendant, Uniden Corporation of America (Uniden), a distributor of electrical equipment including mobile radios, designed a mobile radio that contained software capable of utilizing Johnson's LTR system.[125] In developing the software for the LTR-compatible mobile radios, Uniden disassembled Johnson's LTR software and prepared flow charts for its software using Johnson's LTR software.[126] Johnson subsequently examined Uniden's software and discovered that the software contained in Uniden's mobile units had been copied from Johnson's LTR software.[127] Johnson, consequently, sued Uniden for copyright infringement and sought a preliminary injunction to restrain

[119] *SAS,* 605 F. Supp. at 826.

[120] 623 F. Supp. 1485 (D. Minn. 1985); *see* Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1246 (3d Cir. 1986); (involving copyright protection of nonliteral elements of computer program); SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985) (same).

[121] *Johnson,* 623 F. Supp. at 1497.

[122] *Id.* at 1489.

[123] *Id.* Johnson loaded its software into a Read-Only Memory (ROM) microchip. *Id.*. The mobile radios use an Intel 8049 microprocessor. *Id.* A two-way communication system employing the trunked LTR software is more efficient than traditional two-way communication systems because the software creates an uninterrupted stream of communications. *Id.* The computer software creates the uninterrupted stream by pooling the radio frequencies and making all radio channels accessible to every system user. *Id.*

[124] *Id.* at 1488. Registration of a copyright is prima facie evidence of valid ownership in a copyright. *Id.* at 1492; Novelty Textile Mills, Inc. v. Joan Fabrics Corp., 558 F.2d 1090, 1092 n.1 (2d Cir. 1977); 17 U.S.C. § 410(c) (1982).

[125] *Johnson,* 623 F. Supp. at 1489.

[126] *Id.* at 1490. Disassembly is a process involving the translation of the machine readable object code into source code. *Id.* SIPPL, *supra* note 52, at 115 (disassembly is retranslation of machine language into mnemonics).

[127] *Johnson,* 623 F. Supp. at 1490. In *Johnson,* Johnson's engineers scrutinized one of Uniden's compatible radios at Johnson's lab. *Id.* The Johnson engineers removed the Erasable Programmable Read-Only Memory microchip (EPROM) from Uniden's radio, copied (dumped) the machine language stored on the microchip into a computer, and disassembled the object code. *Id.* Johnson discovered that Uniden's program was extremely similar to Johnson's program. *Id.*

Uniden from selling or marketing Uniden's LTR-compatible radio software.[128]

Before the trial court on Johnson's motion for a preliminary injunction, Uniden attempted to show that the Johnson and Uniden LTR programs were dissimilar by introducing an exhibit consisting of a line-by-line comparison of the Johnson and Uniden codes.[129] The *Johnson* court noted that the machine-languages utilized by the two microprocessors were different because the Johnson radios employed an Intel microprocessor while the Uniden radios used an Hitachi microprocessor.[130] In dismissing the probative value of Uniden's exhibit, which consisted of an attempt to convert the Intel code into the Hitachi code and to compare the two LTR programs on a line-by-line basis, the *Johnson* court found that such a comparison was meaningless because differences in the machine languages made a literal translation of the programs virtually impossible.[131] Rather than considering the two LTR programs on a line-by-line basis, the *Johnson* court focused on the overall structure and design of the programs.[132] Finding a remarkable similarity in the overall design and structure of the two programs and several examples of direct copying, the *Johnson* court concluded that the programs were substantially similar.[133] To support its finding of substantial similarity, the *Johnson* court noted that Uniden's program contained identical error detection tables, the same superfluous instructions, several identical subroutines, a virtually exact paraphrase of Johnson's code, and a strong similarity in the overall design of the Uniden and Johnson LTR programs.[134]

[128] *Id.* at 1497.

[129] *Id.* at 1497-98.

[130] *Id.* at 1497. The district court in *Johnson* noted that a given step in either the Johnson or Uniden program would require a different number of commands depending on whether the program used the Intel or Hitachi microprocessor language. *Id.*

[131] *See id.* at 1497-98 (citing Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 609 F. Supp. 1307 (E.D. Pa. 1985)). The *Johnson* court noted that the literal translation of a program from one microprocessor language to another would be extremely difficult, if not impossible. *Id.* The *Johnson* court also found Uniden's line-by-line comparison unconvincing because Uniden's witness admitted that the translation represented in Uniden's exhibit would not "run" without further modification and that disassembled versions of two programs from different machines would not demonstrate a line-by-line correlation. *Id.*

[132] *See id.* at 1493-97 (recognizing marked similarity in overall design of program and finding that programs were substantially similar).

[133] *Id.* at 1497.

[134] *Id.* The district court in *Johnson* found especially probative of substantial similarity the existence of identical errors, superfluous instructions, the identity of 38 out of 44 subroutines, and the "marked similarity in overall design." *Id.; see* Williams Elec., Inc. v. Arctic Int'l, 685 F.2d 870, 876 (3d Cir. 1982) (existence of identical errors in copyrighted program and contested work is evidence of copying); SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 824 (M.D. Tenn. 1985) (presence of nonfunctional feature in contested program is strong evidence that defendant copied the functionless instruction from plaintiff's program).

Addressing Uniden's defense that Uniden only copied the idea of a mobile radio compatible with the LTR-system, the *Johnson* court noted that idea-expression dichotomy in computer software depends on the single inquiry whether programmers can write other programs that perform the identical function as the copyrighted program.[135] The *Johnson* court noted that the translation, flow charting, and analyzing of a competitor's code in itself does not constitute an infringement.[136] The parties in *Johnson* agreed that translating and analyzing a competitor's code was a standard practice in the computer software industry.[137] The court noted that if Uniden had surveyed only a general outline of the Johnson program and had created detailed code through its own independent imagination and creativity, then Johnson would not have a copyright infringement claim.[138] Accordingly, the *Johnson* court found that the Johnson program was protectible expression because Johnson's code was not the "only and essential means" of creating LTR-compatible software.[139] The *Johnson* court thus concluded that Uniden copied the expression in Johnson's LTR program because Uniden independently could have created an LTR compatible mobile radio without duplication of the Johnson program.[140]

Although the *Whelan, SAS,* and *Johnson* courts recognized that copyright protects the structure and organization of a computer program,[141] the United States District Court for the Southern District of New York in *Q-Co Industries, Inc. v. Hoffman*[142] articulated a narrower scope of copyright

[135] *Johnson*, 623 F. Supp. at 1502 (citing Apple Computer, Inc. v. Formula Int'l Inc., 725 F.2d 521, 525 (9th Cir. 1984)); Apple Computer Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1251 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984)).

[136] *Johnson*, 623 F. Supp. at 1501-02 n.17.

[137] *Id.*

[138] *Id. Compare* Synercom Technology v. University Computing Co., 462 F. Supp. 1003, 1013 n.5, (N.D. Tex. 1978) (preparation of computer program in any programming language from general description'of program is not copy of program, but translation from detailed description of problem solution, such as flowchart or step-by-step set of instructions is copy of program) *with* SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985) (taking copyrighted program, duplicating organization of program in outline form, assigning portions of program to programmers for independent creation, combined with access to source code could constitute copyright infringement).

[139] *Johnson*, 623 F. Supp. at 1501-03. In *Johnson*, the district court found that except for one small aspect of Johnson's program, the "Barker code" essential for establishing communications, Uniden could have created independently a program compatible with Johnson's LTR system. *Id.* at 1503.

[140] *Id.* at 1503.

[141] *See* Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1246 (3d Cir. 1986) (holding that copyright protection for computer programs extends to program's structure, sequence, and organization); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1493-97 (D. Minn. 1985) (recognizing copyright protection for nonliteral elements of computer programs); SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985) (same).

[142] 625 F. Supp. 608 (S.D.N.Y. 1985).

protection for computer programs.[143] In *Q-Co*, the plaintiff, Q-Co Industries, Inc. (Q-Co), was a corporation that provided prompting equipment and services to various customers in business, entertainment, and government.[144] Q-Co marketed prompting software packages designed for use on Atari computers.[145] Q-Co had hired Sidney Hoffman to design a teleprompting program for Q-Co, later entitled the VPS-500.[146] At Hoffman's request, Q-Co had hired Dilip Som to aid Hoffman in developing the VPS-500.[147] In December 1984, Som had begun working on the development of a prompting program for the IBM PC.[148] By the end of December 1984, Hoffman and Som had finished the VPS-500, and Hoffman voluntarily terminated his employment with Q-Co.[149] In January 1985, Hoffman incorporated Computer Prompting Corporation (CPC), and Som and Hoffman became vice-presidents of CPC with both men owning half of CPC's stock.[150] CPC introduced its CPC-1000 teleprompting program for the IBM PC in April 1985, and in June 1985 Q-Co filed suit against Hoffman, Som, and CPC for copyright infringement of Q-Co's VPS-500 program.[151]

[143] *Id.* at 616.

[144] *Id.* at 610. In *Q-Co,* the plaintiff, Q-Co Industries, Inc. (Q-Co) provided customers with teleprompters or prompters, which are machines that scroll text in front of a speaker allowing the speaker to look straight ahead while reading text and thus avoid the appearance of reading. *Id.*

[145] *Id.* at 610-11.

[146] *Id.* at 611. In *Q-Co,* Q-Co hired Hoffman as an employee at will. *Id.* As an employee of Q-Co, Hoffman did not enter into any employment agreement, confidentiality agreement, or agreement not to compete. *Id.*

[147] *Id.* In *Q-Co,* the district court noted that although Q-Co had hired Som as a consultant, Q-Co mostly was unaware of Som's work because Hoffman had conducted almost all of the communications between Q-Co, Hoffman, and Som. *Id.* at 611-12. As a consultant for Q-Co, Som did not enter into any employment agreement, confidentiality agreement, or agreement not to compete. *Id.* at 611. In developing the prompter program for Q-Co, Hoffman and Som worked closely together and used only Q-Co's supplies and equipment. *Id.* Unknown to either Q-Co or Hoffman, Som employed George Schwenk to work on the VPS-500 and paid Schwenk $1,000 for his services. *Id.* at 612. The *Q-Co* court found that Q-Co's program, the VPS-500, was the joint product of Som and Hoffman. *Id.* at 611.

[148] *Id.* at 612. In *Q-Co,* Som and Hoffman decided to develop a prompter program for the IMB PC. *Id.* The IBM PC had a larger memory capacity than the Atari computer, which made an IBM PC prompter more attractive to customers. *Id.* at 611. Developing a prompter program for the IBM PC, however, was more difficult than developing one for the Atari computer because the Atari computer contained a graphics chip that facilitated the generation and scrolling of characters. *Id.* The IBM PC did not contain a graphics chip but required a lengthy and complicated software program to generate and scroll large characters. *Id.*

[149] *Id.* at 612.

[150] *Id.* at 613. In *Q-Co,* Som engaged the services of Martin Smith to help Som in developing a prompter program for the IBM PC. *Id.* at 612-13. By the end of February 1985, Som had finished his work for Q-Co and was working with Smith on developing the CPC-1000, the prompting program for the IBM PC. *Id.*

[151] *Id.* at 613. In *Q-Co,* Hoffman and Som introduced the CPC-1000 at the National Association of Broadcasters (NAB) in Las Vegas. *Id.* Som and Hoffman prepared a brochure introducing the CPC-1000 at the NAB, and the brochure promoted the new program's ability to operate on the IBM PC. *Id.*

Although the *Q-Co* court found that the design and structure of the VPS-500 and the CPC-1000 programs were similar, the *Q-Co* court found no copyright infringement.[152] Recognizing that the CPC-1000 program had four modules that corresponded functionally to four of the VPS-500's twelve modules,[153] the *Q-Co* court noted that the corresponding modules were similar in structure and organization.[154] While finding that Som was completely familiar with the VPS-500 and had employed the "structure and concept of the VPS-500," the court found no direct evidence that Som had used any VPS materials in creating the CPC-1000.[155] The *Q-Co* court also recognized that literal copying of the VPS-500 program was not possible because the two programs were written in different programming languages.[156] Applying the idea-expression dichotomy test, the *Q-Co* court reasoned that the four CPC-1000 modules would be inherent in any prompting program and, therefore, constituted unprotectible ideas.[157]

In arriving at its conclusion that the CPC-1000 program did not infringe Q-Co's copyright in the VPS-500 program, the *Q-Co* court distinguished the *SAS* case, a case in which the *SAS* court found that creating original source code from the outline of another program could constitute copyright infringement.[158] Although Som utilized the basic outline of the VPS-500 program, in the *Q-Co* court's opinion, Som's use of the outline to write the CPC-1000 code did not constitute copyright infringement.[159] In distinguishing the *SAS* case, the *Q-Co* court noted that the existence and organization of the four modules in the CPC-1000 was inherent in the design of any prompting program.[160] The idea-expression merger doctrine, therefore, would permit Som to copy those elements of the VPS-500 program, like the organization and structure, that were an inherent part of any prompting program.[161] The *Q-Co* court further distinguished the *SAS* case on the basis that "slavish copying" of the VPS-500 program was impossible in the CPC-1000 program because of the differences in hardware.[162] The *Q-Co* court, therefore, held that CPC had not infringed Q-Co's copyright because Q-Co merely had copied unprotectible ideas from the VPS program.[163]

[152] *Id.* at 616.

[153] *Id.* at 614.

[154] *Id.*

[155] *Id.* at 615.

[156] *Id.*

[157] *Id.* at 616.

[158] *Id.*

[159] *Id.; see* SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 830 (M.D. Tenn. 1985) (noting that process in which programmers create original code from outline of copyrighted program would constitute copyright infringement).

[160] *Q-Co,* 625 F. Supp. at 616.

[161] *Id.*

[162] *Id.*

[163] *Id.*

While the *Whelan, SAS, Johnson,* and *Q-Co* courts considered whether copying the nonliteral elements of a computer program constituted copyright infringement, the United States District Court for the District of Massachusetts in *Williams v. Arndt and Harvard Investment Service, Inc.*[164] considered whether copying the English language expression contained in a commodities investment manual by creating a computer program from the detailed steps outlined in the investment manual constituted copyright infringement.[165] In rejecting Arndt's defense that creating source code from the English language contained in the trading manual constituted a new expression of the investment system idea, the *Williams* court found that Arndt merely had translated Williams' English language expression of an investment method into a computer programming language.[166] The *Williams* court, therefore, found that the two contested trading systems, Williams' manuals and Arndt's programs, were substantially similar in overall structure and held that Arndt's

[164] 626 F. Supp. 571 (D. Mass. 1985); *see* Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1246 (3d Cir. 1986) (involving alleged copying of nonliteral elements of computer program); Q-Co Indus., Inc. v. Sidney Hoffman, 625 F. Supp. at 615-16 (same); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1490-97 (D. Minn. 1985) (same); SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985) (same).

[165] *Williams,* 626 F. Supp. at 577. The plaintiff in *Williams,* Larry Williams, an investment advisor, wrote an investment manual entitled the Floor Trader's Method (FTM) that described a step-by-step method for trading in various commodities. *Id.* at 573. Williams was an active commodities trader who had authored several books, articles, and pamphlets on commodities trading. *Id.* Williams also conducted several seminars on commodities trading every year. *Id.* The defendant, George Arndt, created a computer program from a copy of the FTM. *Id.* at 574. Arndt received a draft of the FTM manual that did not have any notice of copyright affixed. *Id.* Arndt received the FTM manual from Jerry Snyder who had obtained Williams' permission to have his copy of the FTM programmed for a computer. *Id.* Williams granted Snyder permission to have the FTM computerized on the condition that Snyder would not disclose the method. *Id.* Snyder had sent Arndt a copy of the FTM and had asked Arndt to create a program based on the FTM manual. *Id.* Arndt never created a computer program to Snyder's satisfaction. *Id.* After Snyder had terminated the business relationship, Arndt proceeded to market and sell a computer program that employed the step-by-step method described in the FTM manual. *Id.* Snyder informed Williams that Arndt was selling the FTM in a computer program. *Id.* Williams' attorney sent Arndt a letter in May 1982 demanding that Arndt discontinue selling Arndt's computer program. *Id.* at 575. In March 1982 Williams' attorney sent Arndt another letter advising Arndt that Williams would take legal action if Arndt continued to sell his program. *Id.* On November 1, 1983, Williams sued Arndt for copyright infringement. *Id.* at 576. At trial, Arndt claimed that the source code he devised constituted a "new and different expression of the idea of a market trading system." *Id.* at 576-77. In finding Arndt had infringed Williams' copyright, the *Williams* court reasoned that if copyright law did not protect expressions translated into source code, then every expression would be vulnerable to copying into source code form. *Id.* at 578.

[166] *Id.* at 577-78. The district court in *Williams* found that for a skilled programmer, the conversion of known input, output, and the steps necessary to produce the output is a mere clerical function, which does not involve the new expression of an idea. *Id.* at 577. *See* In the Matter of the Application of John W.C. Sherwood, 613 F.2d 809, 816-17 n.6 (C.C.P.A. 1980) (conversion of known input and known output into computer language is merely clerical function), *cert. denied,* 450 U.S. 994 (1981).

program infringed Williams' copyright in the investment manual.[167]

The courts in *Whelan, SAS, Johnson,* and *Williams* extend copyright protection beyond the literal code of a computer program.[168] In extending copyright protection beyond the source and object code of computer programs, these cases establish that copying the expression in a program, whether embodied in computer programming language, English, or other symbolic representation, constitutes a copyright infringement.[169] *Whelan, SAS, Williams,* and *Johnson* recognize in varying degrees that copyright protection extends to the structure and organization of computer programs.[170] By striking a balance between protecting the monetary incentives for program developers and the free dissemination of ideas, courts have established a standard to distinguish between the protectible expression and unprotectible ideas in the nonliteral elements of a computer program.[171] The established standard is whether the particular structure and organization of a program is the only and essential means or necessary to achieve a desired result.[172]

The deceptively simple idea-expression standard sometimes is difficult to apply. The *SAS* court, for example, suggested that creating source code from

[167] *Id.* at 576, 579. The district court in *Williams* found that Williams' manual and Arndt's programs were substantially similar, even though Williams had written the FTM manual in English and Arndt had written his computer program in source code. *Id.*

[168] *See* Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222, 1246 (3d Cir. 1986) (holding that copyright protection for computer programs extends to program's structure, sequence, and organization); Williams v. Arndt, 626 F. Supp. 571, 577-78 (D. Mass. 1985) (finding that mere translation of commodities trading method from English into computer programming language constitutes copyright infringements); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1493-979 (D. Minn. 1985) (recognizing copyright protection for nonliteral elements of computer program); SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985) (same); *supra* notes 77-140, 164-67 and accompanying text (discussing cases that extend copyright protection beyond literal code of program).

[169] *See* Williams v. Arndt, 626 F. Supp. 571, 577-78 (D. Mass. 1985) (copyright protects expression of idea in detailed English instructions from unauthorized translation or conversion into computer programming language); Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 609 F. Supp. 1307, 1320-21 (E.D. Pa. 1985) (copyright protects expression of idea embodied in computer program even though programmer must adapt or modify expression to run on computers that require different source codes).

[170] *See supra* note 168 (discussing *Johnson, SAS, Whelan,* and *Williams* cases).

[171] *See infra* note 172 (discussing cases that have established standard to distinguish between protectible expression and unprotectible ideas in computer program).

[172] *See* Whelan Assoc., Inc. v. Jaslow Laboratory, Inc., 797 F.2d 1222, 1236 (3d Cir. 1986) (everything that is not necessary to program's purpose or function is protectible expression of idea); M. Kramer Mfg. Co., Inc. v. Andrews, 783 F.2d 421, 436 (4th Cir. 1985) (if only one way to express idea exists, idea and expression merge and there is no copyrightable material); Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521, 525 (9th Cir. 1984) (noting that if programmers can write other programs that perform same function as particular program, then program is expression of idea); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983) (noting that no merger of idea and expression occurs when other methods for expressing idea are not foreclosed as practical matter), *cert. dismissed,* 464 U.S. 1033 (1984); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1502 (D.

an outline of a competitor's source code would constitute an appropriation of protectible expression.[173] The *Johnson* court, however, noted that the independent creation of source code from the general outline of a competitor's program would not constitute copyright infringement.[174] The *SAS* and *Johnson* courts' positions are not inconsistent necessarily. If an outline is general enough to represent a basic structure necessary to achieve a desired result, then using such an outline to create source code would not result in copyright infringement.[175] If, however, the outline represented only one of innumerable ways to organize and structure a computer program, then the outline would embody protectible expression.[176] Consequently, if a programmer creates source code that embodies the outline's detailed structure and organization, such creation would constitute copyright infringement.[177]

The *Q-Co* case illustrates the difficulty of applying the idea-expression standard to a particular set of facts.[178] The *Q-Co* court, after concluding that the structure and organization of the contested programs were similar, determined that the similarity did not constitute copyright infringement because the structure and organization of the modules in Q-Co's program would be inherent in any prompter software.[179] The *Q-Co* court, however, arguably misapplied the idea-expression doctrine.[180] The *Q-Co* court dismissed the importance of the similarities in structure by stating that differences in hardware made structural copying impossible.[181] Although the district court in *Q-Co* noted that the CPC-1000 was not a paraphrase of the VPS-500, the *Q-Co* court refused to recognize that programmers can appropriate

Minn. 1985) (idea-expression dichotomy test involves single inquiry of whether programmers can write independently other programs that perform same function as copyrighted program); CONTU FINAL REPORT, *supra* note 4, at 20 (idea-expression distinction in computer context means that when specific instructions, though previously copyrighted, are "only and essential means" of accomplishing task, then later use of those instructions by another programmer is not copyright infringement); *supra* notes 35-45 and accompanying text (discussing merger of idea and expression in computer software).

[173] *SAS,* 605 F. Supp. at 826.

[174] *Johnson,* 623 F. Supp. at 1501 n.17.

[175] *See supra* note 172 (explaining that when particular work represents only and essential means of achieving result, idea and expression have merged, and work represents unprotectible idea).

[176] *See supra* note 38 (explaining expression of idea in computer software).

[177] *See supra* notes 35-41, 172 and accompanying text (discussing distinction between idea and expression in computer software).

[178] *See* Q-Co Indus., Inc. v. Hoffman, 625 F. Supp. 608 (S.D.N.Y. 1985) (court applying idea-expression distinction analysis).

[179] *Id.* at 614-16.

[180] *See id.* (recognizing that copyright extends only to forms of expression and not ideas, but arguably misapplying test for determining what constitutes expression of idea in computer program); *infra* notes 181-189 and accompanying text (discussing how *Q-Co* court arguably misapplied idea-expression dichotomy doctrine); *supra* notes 35-45, 171-72 and accompanying text (discussing idea-expression dichotomy in computer software).

[181] *Id.* at 616.

a program's protectible expression even though a programmer modifies or adapts the program to operate on a different computer.[182] *Whelan, SAS*, and *Johnson*, however, suggest that such nonliteral copying of a program, known as the "conversion" of a program, constitutes copyright infringement.[183]

According to Q-Co's expert, Dr. Friedberg, the close correspondence between the four modules, including similarities in the structure, organization, and a few textual similarities, was surprising except when viewed in the context of conversion.[184] Although the four modules were a necessary design feature for the VPS-500 because of the memory limitations on the Atari computer, the IBM PC did not have the same memory limitations and, therefore, did not require the four-module design.[185] According to Dr. Friedberg, the only logical explanation for the four-module design was CPC's conversion of the Q-Co program.[186] Dr. Friedberg's testimony, therefore, casts doubt on the court's conclusion that the four modules were "inherent" in the design of any prompting program.[187]

The evidence of similarities outlined by the *Q-Co* court suggests that the structure and organization of the four modules in the VPS-500 program constituted protectible expression because these elements were not the "only and essential means" of creating a prompter program.[188] The *Q-Co* court, therefore, arguably determined incorrectly that CPC merely copied the idea of the VPS-500.[189]

While the *Whelan, SAS, Williams,* and *Johnson* courts recognize the copyright protection of the structure and organization of computer pro-

[182] *Id.* at 615-16.

[183] *See* E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1497 (D. Minn. 1985) (concluding that Uniden's conversion of Johnson's LTR software to operate on Hitachi microprocessor constituted copyright infringement); SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816, 829-30 (M.D. Tenn. 1985) (concluding that S & H's conversion of SAS' program constituted copyright infringement); Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 609 F. Supp. 1307, 1320-21 (E.D. Penn. 1985) (concluding that Jaslow's conversion of Dentalab program, which operates on IBM Series/1 computer, to operate on IBM-PC constituted copyright infringement), *aff'd*, 797 F.2d 1222, 1246 (3d Cir. 1986).

[184] *Q-Co,* 625 F. Supp. at 614.

[185] *Id.*

[186] *Id.*

[187] *See Q-Co,* 625 F. Supp. at 614 (discussing testimony of Q-Co's expert witness).

[188] *See id.* at 613-15 (discussing similarities between CPC-1000 and VPS-500 programs); Whelan Assoc., Inc. v. Jaslow Dental Laboratories, Inc., 797 F.2d 1222, 1236 (3d Cir. 1986) (everything not necessary to purpose or function of program is part of program's expression of idea); Apple Computer, Inc. v. Formula Int'l., Inc., 725 F.2d 521, 525 (9th Cir. 1984) (if programmers can write other programs that perform same function as particular program, then program is expression of idea); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1502 (D. Minn. 1985) (idea-expression dichotomy test involves single inquiry of whether programmers can write independently other programs that perform same function as copyrighted program); CONTU FINAL REPORT, *supra* note 4, at 20 (idea-expression distinction in computer context means that when specific instructions though previously copyrighted are "only and essential means" of accomplishing task, then later use by another is not infringement).

[189] *See Q-Co,* 625 F. Supp. at 616 (holding that defendants merely copied idea of Q-Co's VPS-500 program).

grams,[190] these cases are not as clear about whether the user interface of a program constitutes protectible expression under copyright law. Despite the uncertainty surrounding the protectibility of the user interface through the copyright laws, the user interface of a commercial software package is a valuable element of the program.[191] In at least one case, a federal district court has recognized that the user interface comprised an important element in the design of user-oriented programs.[192] In *Dickerman Associates v. Tiverton Bottled Gas Co.*,[193] the United States District Court for the District of Massachusetts decided a case involving the trade secrecy protection of a program designed to provide accounting and management information to independent petroleum jobbers.[194] In language pertinent to the copyright protection of a program's user interface, the court noted that the visual screens demonstrated numerous decisions in the design of the program.[195] Recognizing that the visual displays of any accounting program would include certain basic information, the *Dickerman* court acknowledged that the screen would not have to include other options, like certain financial data.[196] The *Dickerman* court also noted that the programmers made design decisions concerning the arrangements of menus and submenus, the way in which users would access submenus, how many submenus each function should include, and how to organize the information on each screen.[197] The *Dickerman* court

[190] *See supra* note 168 (discussing *Johnson, SAS, Whelan* and *Williams* cases).

[191] *See supra* note 67 (noting that user interface can be most valuable element of user-oriented program).

[192] *See* Dickerman Assoc., Inc. v. Tiverton Bottled Gas Co., 222 U.S.P.Q. (BNA) 529, 530-31 (D. Mass. 1984) (recognizing user interface as valuable element in design of computer program).

[193] 222 U.S.P.Q. (BNA) 529 (D. Mass. 1984).

[194] *Id.*

[195] *Id.* at 530-31. The court in *Dickerman* recognized that the user interface is an important element in the organization of a computer program:

> The JMS program starts with a "Systems Options Menu," the main menu, which includes five major and five minor groupings of functional options . . . The major groupings determine the organization of the system and were chosen by the designers . . . from an infinite number of alternatives . . .
>
> The customer display screen demonstrates numerous design decisions. While the inclusion of certain information, name, address, billing and payment data is self evident, it is less imperative that this screen should for example show year to date of delivery and payment data, or that it be arranged in the precise manner of the JMS.
>
> In addition to choosing the particular five major groupings . . . the designers thus made decisions concerning the specific manner in which these submenus are to be used, how they are to be accessed, how many screens to use, and how to arrange the information on each screen. It is the organization of the JMS program, with its five major groupings, combined with the particular features within the system and the procedures to be employed in its use, which plaintiff claims as its trade secret.

Id.

[196] *Id.* at 530.

[197] *Id.*

recognized that the user interface is an important design element for the organization of a computer program.[198]

Since recent cases have extended copyright protection to the structure and organization of computer programs, courts should extend copyright protection to the user interface because the user interface is an important element of authorship incorporated in the structure and organization of a computer program.[199] The same idea-expression principles would apply to the user interface as in other aspects of nonliteral copying.[200] Copyright law should not protect those elements of the usèr interface that are inherent in the idea of the program.[201] Copyright law, however, should protect those elements of the user interface that reflect a programmer's design decisions in creating the particular expression of an idea.[202]

The copyright protection of traditional literary works supports the *Whelan, SAS, Williams,* and *Johnson* courts' decisions to extend the copyright protection beyond the literal expression in computer programs.[203] Judge Learned Hand, when enunciating his celebrated "abstractions test," established that the protection of more than the literal text is essential to protect literary property.[204] In his abstractions test, Learned Hand suggested that when a plagiarist does not copy a work verbatim, but copies an abstract of the entire work, the determination of when the plagiarist has appropriated the work is more difficult.[205] Judge Hand suggests that literary works fit into an increasing number of patterns of greater generality as one extracts more of the details from a particular work.[206] At some point in the levels of abstraction, the idea and the expression of the idea will become inseparable.[207] At this level of abstraction, copyright will not protect duplication of the work.[208] Judge Hand and other commentators have recognized that limiting protection to the literal text of a work would vitiate the economic motivations underlying copyright law.[209] Commentators have noted that copyright law

[198] *Id.; see supra* note 195 and accompanying text (discussing that major groupings in program's menu can determine program's organization).

[199] *See supra* notes 77-140, 164-67 and accompanying text (discussing cases recognizing copyright protection for nonliteral elements of computer program).

[200] *See supra* notes 35-45 and accompanying text (discussing idea-expression dichotomy test for computer software).

[201] *See supra* notes 35-41 and accompanying text (discussing principle that copyright does not protect expression which has merged with idea).

[202] *See supra* notes 171-72 and accompanying text (copyright protects expression in computer programs when expression is not only and essential means of achieving idea of program).

[203] *See supra* note 170 (discussing *Johnson, SAS, Whelan,* and *Williams* cases).

[204] *See* Nichols v. Universal Pictures Corp., 45 F. 2d 119, 121 (2d Cir. 1930) (effective protection of literary works demands that copyright protect nonliteral copying of literary works).

[205] *Id.*

[206] *Id.*

[207] Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1960).

[208] *Id.*

[209] *Id.*; 1 NIMMER, *On Copyright* § 1.10 of [B] at 1-72, -73 (1985); *supra* notes 13-15 and accompanying text (discussing policy of copyright law).

should protect an author's particular selection, arrangement, and sequence of ideas, as well as the author's literal text.[210]

In addressing the issue of nonliteral copying of traditional works, courts also have held that copyright protection extends to the structure and organization of literary works that are analogous to computer programs such as instruction manuals and textbooks.[211] In *Meredith Corp. v. Harper & Row Publishers,*[212] for example, the United States District Court for the Southern District of New York considered whether the copying of the selection and organization of a text on child psychology constituted copyright infringement.[213] The *Meredith* court found that although the defendant had added some differing structure in creating the arrangement and selection of its textbook, the structure and organization were primarily the result of the defendant's copying of an outline made from plaintiff's textbook.[214] Accordingly, the *Meredith* court found that the defendant had infringed the copyright of the plaintiff's book.[215]

Further support for the extension of copyright protection to the nonliteral elements of a computer program is found in *Orgel v. Clark Boardman Co.*[216] In *Orgel,* the Second Circuit noted that the appropriation of another's creative and intellectual labors in publishing a competing work without

[210] *See* NIMMER, *supra* note 209, at 1-73 to 1-74 (copyright law should protect author's selection and arrangement of ideas); CHAFEE, *Reflections On the Law of Copyright: I,* 45 COLUM. L. REV. 503, 513-14 (1945) (copyright law should protect pattern of work and sequence of events in work); Brief of Amicus Curiae at 22, Whelan Assoc., Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986) (copyright law should protect programmer's selection and arrangement of ideas by extending protection to structure, sequence, pattern, and organization of computer programs).

[211] *See* Meredith Corp. v. Harper & Row, Publishers, Inc., 378 F. Supp. 686, 690 (S.D.N.Y.), *aff'd,* 500 F.2d 1221 (2d Cir. 1974); *permanent injunction entered,* 413 F. Supp. 385 (S.D.N.Y. 1975) (finding defendant guilty of copyright infringement when defendant had copied directly only small portion of plaintiff's textbook but had copied extensively structure and topical sequence of plaintiff's work); *see also* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930) (effective protection of literary works demands that copyright protect literary works from nonliteral plagiarism); Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49, 55 (2d Cir.) (person can appropriate expression in play without using play's dialogue), *cert. denied,* 298 U.S. 669 (1936).

[212] Meredith Corp. v. Harper & Row, Publishers, Inc., 378 F. Supp. 686-690 (S.D.N.Y), *aff'd,* 500 F.2d 1221 (2d Cir. 1974); *permanent injunction entered,* 413 F. Supp. 385 (S.D.N.Y. 1975).

[213] *Meredith,* 378 F. Supp. at 690.

[214] *Id.* In *Meredith Corp. v. Harper & Row Publishers,* Meredith Corporation (Meredith) decided to publish a textbook on child psychology. *Id.* at 687. Meredith conducted a market survey and determined that a textbook, authored by Drs. Mussen and Kagan, was a leading book in the field. *Id.* Consequently, Meredith decided to use the Mussen book as a model for the selection, weight, and sequencing of topics. *Id.* Meredith employees prepared detailed outlines of the chapters in the Mussen book and distributed the outlines to freelance writers to prepare the text of Meredith's textbook. *Id.* The freelance writers that Meredith employed often had no background in psychology. *Id.*

[215] *Id.* at 690. The court in *Meredith Corp. v. Harper & Row Publishers* noted that a scholar like Dr. Mussen, who coauthored the textbook published by Harper & Row Publishers, Inc., would have no incentive to do the years of research and scholarship necessary to create such a textbook when others could merely paraphrase Dr. Mussen's work and compete with him. *Id.*

[216] 301 F.2d 119 (2d Cir.), *cert. denied,* 371 U.S. 817 (1962).

expending the same time and effort necessary to create independently the same result is a copyright infringement.[217] The copying of the structure and organization of computer programs, as the *SAS* court clearly articulated, constitutes the appropriation of significant creative and intellectual labors.[218] If copyright law did not protect the structure and organization of computer programs, competitors of a software developer would receive a windfall by saving substantial resources in developing a successful structure and organization for rival programs.[219] Consequently, the failure of copyright law to protect these elements of a program would vitiate software developers' monetary incentives, which copyright law strives to protect.[220]

Following the lead of the *Whelan, SAS, Johnson* and *Williams* courts,[221] the federal judiciary should continue to expand the scope of protection for computer programs to encompass programs' user interface, structure, and organization, and thereby promote Congress' desire to foster creative and intellectual achievements in computer programming through the copyrights laws.[222] The judicial extension of copyright protection to these elements of computer programs is consistent with the historical application of copyright law to new technologies.[223] Just as Congress recognized the need to protect the valuable rights in motion pictures and sound recordings, courts also should extend copyright protection to encompass the valuable rights in the user interface, structure, and organization of computer programs.[224] Because copyright law classifies computer programs as literary works, the copyright protection of the structure and organization in traditional literary works likewise compels the protection of these elements in computer programs.[225] In

[217] *Id.* at 120.

[218] *See* SAS Inst., Inc. v. S & H Computer Sys., 605 F. Supp. 816, 826 (M.D. Tenn. 1985) (appropriation of completed and proven design of computer program saves competitors substantial time and effort in developing original design); *supra* notes 62-71 and accompanying text (discussing importance of structure and organization in development of computer programs).

[219] SAS Inst., Inc. v. S & H Computer Sys., 605 F. Supp. 816, 826 (M.D. Tenn. 1985).

[220] *See* Brief of Amicus Curiae at 6, Whelan Assoc., Inc. v. Jaslow Dental Laboratory, 797 F.2d 1222 (3d Cir. 1986) (limiting copyright protection to literal text would inhibit dissemination of new programs to detriment of society and computer industry); *supra* note 10 and accompanying text (discussing how copyright law protects programmers' monetary incentives to promote creation of new programs).

[221] *See supra* note 168 (discussing *Johnson*, SAS, *Whelan* and *Williams* cases).

[222] *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1254 (3d Cir. 1983) ('80 amendments to '76 Act reflect Congress' receptivity to new technology and Congress' desire to foster, through copyright laws, continued creativity and imagination in computer software), *cert. dismissed,* 464 U.S. 1033 (1984); *supra* notes 13-15 and accompanying text (discussing policy of copyright law).

[223] *See supra* note 16 (discussing historical application of copyright law to new technologies).

[224] *See supra* note 16 (discussing extension of copyright law to protect valuable rights in motion pictures and sound recordings).

[225] *See supra* note 22 (discussing classification of computer programs as literary works

determining whether to extend protection to the structure and organization in a particular program, courts must balance the need for protection of the programmer's creative labors with society's need for the free dissemination of ideas.[226] Whenever a copyrighted program consists of protectible expression, so that other programmers are able to create a computer program that performs the same function as the copyrighted program through the independent creation of the structure, organization, and user interface, courts should protect the copyright owner's valuable interests in these elements of the program.

DANIEL J. FETTERMAN

under copyright law); *supra* notes 203-15 and accompanying text (discussing copyright protection of nonliteral elements in traditional literary works).

[226] *See supra* notes 13-15 and accompanying text (explaining fundamental policy of copyright law as balance between encouraging creation of new works by protecting author's economic incentives and protecting society's interest in free dissemination of ideas).