Rutgers Computer and Technology Law Journal

1995

21 Rutgers Computer & Tech. L.J. 88

Copyright Infringement of Protected Computer Software: An Analytical Method to Determine Substantial Similarity

NAME: Donald F. McGahn II *

BIO:
* B.A., 1991, University of Notre Dame; J.D., 1994, Widener University School of Law; Member, New Jersey and Pennsylvania Bars; Judicial Clerk for the Honorable Charles R. Alexander, Court of Common Pleas, 18th Judicial District, Clarion, Pennsylvania. The author expresses his gratitude to Professor Gregory Gelfand of the Widener University School of Law for his helpful critique on an earlier draft of this article. The author would also like to thank David Holderer, a systems engineer at Electronic Data Systems, and Daniel McGahn, a Masters of Engineering candidate at the Massachusetts Institute of Technology, for their invaluable technical advice.

LEXISNEXIS SUMMARY:
... The coming decade will be a pivotal period in the development of this computer revolution. ... In short, copyright protection extends only to the expression of ideas not the ideas themselves. ... Under the copyright law doctrine of merger, a close cousin to the idea/expression dichotomy, copyright protection will be denied to even some expressions of ideas if the idea behind the expression is such that it can be expressed only in a very limited number of ways. ... However, in Whelan Associates v. Jaslow Dental Laboratory, Inc., the Third Circuit rejected Synercom, and focused instead on the "idea/expression" dichotomy in order to extend copyright protection to the program's structure, sequence, and organization. ... For example, Professor Melville B. Nimmer reasoned that "the crucial flaw in ... [the Whelan test's reasoning] is that it assumes that only one "idea,' in copyright law terms, underlies any computer program, and that once a separable idea can be identified, everything else must be expression." ... If the fact-finder finds that the ideas are sufficiently similar, then substantial similarity of the expressions must be determined; the "ordinary reasonable person" standard is used when making a determination concerning similarity. ... In computer programs, especially multi-task

software, it is not uncommon to find a plethora of ideas. ...

TEXT:
[*89]

I. Introduction

There is a revolution underway, whose scope will exceed that brought about by the invention of printing. This revolution will come about because of the advent of cheap and powerful logical devices ... Now is the time to tune in to the current. It will soon be all around you. The coming decade will be a pivotal period in the development of this computer revolution. The primary phenomenon of the next decade, simply put, will be the advent of widespread computer literacy ... But the most persuasive change will be that computers will leave the realm of the academic, military, and business elites and become tools and companions of the everyday educated person. [n1]

When Joseph Deken made this statement in 1981, few could [*90] have appreciated his accuracy and foresight. At that time, computers were extremely technical and expensive room-sized machines consisting of complex mazes of wires, switches, and vacuum tubes, and were available to only a handful of highly trained and well-funded commercial programmers. Today, however, few would dispute Deken's prophecy, as the growth of the personal computer industry over the past ten years has been astronomical. [n2] Coupled with this growth is a rise in computer literacy, as evidenced by the increased sales of computer software. [n3]

Although many predicted this expansion, Congress has consistently remained several steps behind [n4] in protecting what is fast becoming a substantial part of the United States' economy. [n5] Because the law is vague, [n6] courts have been forced to fashion [*91] standards for determining when copyright infringement of protected computer software occurs. [n7] Some have attempted to apply the traditional "idea/expression" dichotomy [n8] to computer programs. The results have been mixed. [n9] This inconsistency is not surprising, since judges are not schooled in computer programming, but in the analytical underpinnings of the law. The result is a variety of judicially created tests, which rely on amorphous terms such as "ordinary observer," [n10] "extrinsic/intrinsic," [n11] "total concept and feel," [n12] and "iterative." [n13]

This article will develop an analytical method to determine whether copyright infringement of software has occurred. This three-part test, based largely on Learned Hand's abstractions test, [*92] allows courts to rely on well-established and familiar principles of copyright law, yet, at the same time, it provides the predictable results necessary to foster innovation and development within the computer industry. The first prong requires the application of Judge Hand's classic abstraction formula in order to separate unprotectable ideas from protectable expression. Second, after disregarding the unprotectable idea, unprotectable expression is filtered out by utilizing principles such as the merger doctrine. Finally, the portions which remain are compared, and a substantial similarity determination is made.

This article begins with an introduction to computer systems, software, and terminology. The article then reviews the history of copyright law, paying particular attention to those changes in the statutory law that were targeted at the computer industry. This will demonstrate how the analytical method is an adaptation of already existing legal principles. Third, various tests which courts have developed will be introduced, and then criticized for their inadequacies. The three-part analytical method is then presented. The article concludes with the application of this analytical method, and a consideration of potential criticism that may be lodged against it.

II. Background

A. Introduction to Computer Systems, Software, and Terminology.

A computer system is composed of two distinct types of components - hardware and software. [n14] Hardware encompasses the physical components of the computer system. These include, among other things, the central processing unit (CPU), keyboard, monitor, disk drives and memory. [n15] Software, on the other hand, refers to computer programs, which are the set of programming instructions that control the hardware. [n16] Two categories [*93] of software exist - operating systems and applications. [n17] An operating system controls the operation of the entire computer system, including, for instance, data storage and printing. [n18] Of course, different computers contain different operating systems, such as DOS and OS/2. Application programs, in contrast, are designed to enable the user to perform certain tasks. [n19] Word processing, video games, spreadsheets, and databases are some common examples. [n20] Computer programs such as WordPerfect, dBase, and Lotus 123 are examples of common applications.

1. The Three Different Levels of Computer Languages.

Programmers use computer languages to write computer programs. These languages have three levels: high level, assembly, and machine. [n21] High level languages include BASIC, COBOL, C, PASCAL, and FORTRAN. Because they most closely resemble the English language, high level languages are the most commonly used and are the easiest to comprehend. [n22] Assembly language, which uses alphanumeric labels, although more difficult to understand, [n23] is nonetheless highly efficient. [n24] [*94]

2. The Difference Between Source and Object Code.

Programs in source code include those written in either high level or assembly language. [n25] Source code, in its raw form, cannot be understood by computers, and hence must be translated into a comprehensible language. [n26] A compiler is used to accomplish this translation. [n27] The compiler is nothing more than a program which converts high level source code into object code. [n28] Object code, [n29] commonly referred to as binary code, is comprised of a string of "1" and "0." [n30] Although object code can be understood by a computer, it is nearly impossible for lay people to [*95] comprehend. [n31]

3. The Distinction Between Literal and Non-Literal Elements.

In analyzing the scope of protection, courts have frequently referred to a program's "literal" and "non-literal" parts. [n32] The literal elements generally include the written elements of the program, for instance, the source or object code, [n33] while the non-literal elements are intangible, stylistic elements. [n34] Non-literal elements consist of the structure, sequence, and organization, as well as the other creative aspects of a program. [n35] The screen display is an additional non-literal element. [n36]

B. The Development of Copyright Law.

An examination of the amount of protection afforded computer programs requires an understanding of the fundamental principlesand history underlying copyright law. It is these principles which must be applied to any analysis of the copyright [*96] ability of computer programs. [n37]

1. The Constitutional Basis for Congressional Protection.

Copyright protection is derived directly from the text of the United States Constitution. The Constitution provides that "Congress shall have [the] Power ... to promote the Progress of Science ... by securing for limited Times to Authors and Inventors the exclusive Right to their ... Writings and Discoveries." [n38] Pursuant to this enumerated power, Congress enacted the first copyright law in 1790. [n39] This first act only provided protection for "any map, chart, book or books now printed." [n40] Congress expanded this protection in 1802 to encompass designs, prints, etchings and engravings, [n41] and, in 1856, to include "dramatic composition." [n42] "Photographs and the negatives thereof" were included in 1865. [n43] "Statuary" and "models or designs intended to be perfected as works of fine art" were added in 1870. [n44]

Congress eventually took a more general approach to copyright. In 1909, it replaced the list of specific categories with the following language: "the works for which copyright may be secured [*97] under this title shall include all the writings of an author." [n45] Included was a non-exclusive list designed to explain the meaning of "all the writings of an author." [n46] Congress expanded this list in 1912 to include "motion pictures" [n47] and, in 1972, to include "sound recordings." [n48]

As early as 1955, Congress considered changing the 1909 Act. In 1976, it implemented new legislation. [n49] The 1976 Copyright Act provided protection for "original works of authorship fixed in any tangible medium of expression." [n50] Congress broadly defined this fixation requirement as any medium "now known or later developed, from which they [the works of authorship] can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device." [n51] This medium can take a variety of forms, [n52] including:

> words, numbers, notes, sounds, pictures, or any other graphic or symbolic indicia, whether embodied in a physical object in written, printed, photographic, sculptural, punched, magnetic, or any other stable form, and whether it is capable of perception directly or by means of any machine or device now known or later developed. [n53]

A computer program satisfies this fixation requirement if it is either written on paper or stored in some physical device. [n54] [*98]

2. The "Idea/Expression" Dichotomy.

The dispositive issue in copyright law is the "idea/expression" dichotomy, codified by section 102 of the 1976 Act. [n55] As embodied therein, "any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work," [n56] is unprotectable. In short, copyright protection extends only to the expression of ideas not the ideas themselves. [n57]

3. Merger.

It is often difficult to separate an idea from an expression, particularly since there is only a finite number of ways to express a certain idea. This concept is commonly known as the doctrine of merger. When merger occurs, copyright protection is denied. [n58] [*99] If such protection were afforded, the practical result would be a de facto monopoly over a particular idea. [n59] Such a result would frustrate the purpose behind copyright law which is designed to protect the expression of an idea but only "for a limited time." [n60]

The genesis of the concept of merger lies in the landmark case of Baker v. Selden. [n61] There, the plaintiff, Selden, had obtained a copyright on a book which described a new accounting method. [n62] Contained in the book were blank forms that were to be used in the new system. [n63] Selden argued that Baker's accounting book, which contained copies of Selden's forms, infringed his copyright. [n64] The U.S. Supreme Court considered whether Selden's blank forms were part of the unprotectable method, that is, the idea of his book, or instead were part of the copyrightable text - the expression. [n65]

In resolving this issue, the Court determined:

> Where the art it teaches (i.e., idea) cannot be used without employing the methods and diagrams used to illustrate the book (i.e., expression), or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given to the public. [n66]

Accordingly, the Court held that the blank forms were necessary incidents to Selden's method of accounting, and were therefore not entitled to any copyright protection. [n67]

More recently, the Third Circuit has reasoned:

> The merger principle ... is a variation of the idea/expression dichotomy ... When the idea and the expression of the idea coincide, then the expression will not be protected in order to prevent creation of a monopoly on the underlying "art." ... An expression will be found to be merged into the idea when "there are no or few other ways of expressing a particular [*100] idea." [n68]

The Eighth Circuit has similarly described the concept of merger:

> Under the copyright law doctrine of merger, a close cousin to the idea/expression dichotomy, copyright protection will be denied to even some expressions of ideas if the idea behind the expression is such that it can be expressed only in a very limited number of ways. The doctrine is designed to prevent an author from monopolizing an idea merely by copyrighting a few expressions of it. [n69]

C. The Extension of Copyright Law to Computer Programs.

1. CONTU and the 1980 Amendments.

The Copyright Act of 1976 specifies seven categories of protectable works of authorship. [n70] Although computer programs are not mentioned as a separate category, they are encompassed within the category of "literary works." The original language of Section 117 states:

> Notwithstanding the provisions of sections 106 through 116 and 118, this title does not afford to the owner of copyright in a work any greater or lesser rights with respect to the use of the work in conjunction with automatic systems capable of storing, processing, retrieving, or transferring information, or in conjunction with any similar device, machine, or process, than those afforded to works under the law, whether title 17 or the common law or statutes of a State, in effect on December 31, 1977, as held applicable and construed by a court in an action brought under this title. [n71]

[*101]
Prior to the 1976 Act, the Copyright Office had begun accepting registrations for computer programs. Congress' intent here is clearly to maintain the "status quo." [n72] Evidence of this desire is also found in the legislative history surrounding "literary works," a category which "includes computer data bases, and computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves." [n73]

Although Congress included computer programs in the 1976 Act, it was apparent, even prior to 1976, that further clarification would be necessary. Consequently, Congress created the Commission on New Technological Uses (CONTU) in 1974. [n74] Congress granted CONTU three years to "make recommendations on legislation or procedures" [n75] concerning computer copyright infringement.

CONTU published its final report in 1978. [n76] The Commission concluded that because of the high cost of developing computer programs, such programs should receive protection, in order to allow the creator to spread the costs by producing multiple [*102] protected copies. [n77] CONTU apparently believed, however, that existing copyright protection was sufficient. The Commission noted that

Congress understood that "section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law." n78

CONTU therefore proposed two minor recommendations. First, it recommended that section 101 be amended to define "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." n79 The second recommendation sought to allow the owners [*103] of computer programs, in certain circumstances, to make copies or adaptations of their programs. n80 Congress adopted both recommendations by amendment in 1980. n81

2. The Structure, Sequence, and Organization of Programs.

Because all computer programs are now classified as "literary works," it is no longer relevant whether the program is written in source or object code. Nevertheless, in the wake of the 1980 Amendments, n82 courts have generally held that both source and object codes are entitled to protection. n83 [*104]

In the seminal case of Apple Computer, Inc. v. Franklin Computer Corp., n84 the Third Circuit determined that "a computer program, whether in object code or source code, is a "literary work' and is protected from unauthorized copying, whether from its object or source code version." n85 In so holding, the court disposed of three arguments. n86 Of course, it rejected the proposition that copyright does not protect object code. n87 Unlike source code, object code cannot be "perceived" or "read" simply by looking at the code. Second, Judge Sloviter, writing for the court, rejected the notion that copyright does not protect software embedded in a read only memory (ROM), because a ROM is a machine part or utilitarian object rather than a work of authorship. n88 Finally, Judge Sloviter refused to accept the argument that copyright does not protect operating system programs, as opposed to application programs, because an operating system program is a process, system, or method of operation which is protectable, if at all, under patent laws and not under copyright. n89

Although Apple made it clear that computer programs are protectable by copyright, the scope of copyright protection [*105] remained unresolved. Specifically, it remained unclear whether the structure, sequence, and organization of a computer program was copyrightable. A second wave of cases dealing with the scope of protection was therefore inevitable. n90 In Synercom Technology v. University Computing Co., n91 the court ruled that the sequence of certain input formats was simply an unprotectable idea. n92 The program at issue was a data input sequence used to enter data into [*106] a complex structural analysis computer program. n93 Prior to the design of the sequence, a similar structural analysis program existed, n94 but use of it required extensive and complicated training. n95 However, if the structural analysis program were modified to accept the input sequence in question, the program would be easier to use. n96 The sequence was contained on a series of punch cards that were loaded in a particular order. n97 Judge Higginbotham, in holding that the sequence was not copyrightable, was unable to separate the idea from the expression. He explained:

As noted "in cases of literary or artistic works, and works of similar character, in which the form, arrangement, or combination of ideas represents the product of labor and skilled effort separate and apart from that entailed in the development of the intellectual conception involved,' copyright protection is available. Here if order and sequence is the expression, the skilled effort is not separable, for the form, arrangement, and combination is itself the intellectual conception involved. It would follow that only to the extent the expressions involve stylistic creativity above and beyond the bare expression of sequence and arrangement, should they be protected. [n98]

In SAS Institute v. S & H Computer Systems, [n99] the court ruled that S & H's software, which was written for use on VAX computers manufactured by the Digital Equipment Corporation (DEC), was "substantially similar" [n100] to SAS' product, which was designed for use on IBM equipment. [n101] Furthermore, the S & H software was substantially and pervasively based, in the court's view, upon the SAS product. [n102] Hence, Judge Wiseman found it [*107] to be a derivative work. [n103] In addition, the court refused to characterize forty-four specific instances of copying in the context of approximately 186,000 lines of source code as "insubstantial." [n104] Judge Wiseman also noted copying was, in fact, much more pervasive before S & H started destroying evidence and concealing its conduct. [n105] The court ultimately concluded that the defendant had copied the organizational and structural details of the SAS software, and this aspect of the copying pervaded the entire S & H product. [n106]

However, in Whelan Associates v. Jaslow Dental Laboratory, Inc., [n107] the Third Circuit rejected Synercom, and focused instead on the "idea/expression" dichotomy in order to extend copyright protection to the program's structure, sequence, and organization. [n108] The computer program at issue was designed for a dental laboratory. [n109] The plaintiff had created a program called Dentalab for the defendant [n110] for use on IBM Series One computers. [n111] A dispute later arose as to who, in fact, owned Dentalab. The defendant, meanwhile, had created a similar program called Dentcom which ran on personal computers. [n112] Advertising indicated that Dentcom was "a new version of the Dentalab computer system." [n113] The court decided that, although the defendant did not literally copy plaintiff's program, the defendant nonetheless infringed upon the original program's protected structure, sequence, and organization. [n114]

As proof of the infringement, the "order entry" modules of the two programs were entered into evidence:

ORDER ENTRY PROGRAMS

[SEE TABLES IN ORIGINAL] [*109]

The Third Circuit concluded that the purpose of the program was "to aid in the business operations of a dental laboratory." [n116] Writing for the court, Judge Becker fashioned the following rule: "the purpose or function of a utilitarian work [such as a computer program] would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea." [n117] This purpose then was the "idea" and everything more specific constituted expression. To determine this, the court separated the idea from the expression. [n118] Because the particular sequence of the Dentalab program was only one of several possible sequences, it constituted protectable expression. [n119] [*110] There were, in fact, other such programs on the market which performed the same function but had different structures. [n120]

The Whelan court had to "come to grips" [n121] with Synercom Technology v. University Computing Co. [n122] before extending copyright protection to encompass the sequence, structure, and organization of computer programs. Judge Becker ultimately determined that "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, we think that it is incorrect." [n123] The Copyright Act of 1976, in the Whelan court's view, clearly "demonstrates that Congress intended sequencing and ordering to be protectable in the appropriate circumstances ...." [n124] Computer programs do not fall within an exception to this general rule. [n125]

Both courts and commentators have criticized Whelan. For example, in Computer Associates International, Inc. v. Altai, Inc., [n126] the Second Circuit found the test "simplistic," [n127] "inadequate," and "inaccurate." [n128] The court refused to apply the similarity test in the context of computer programs. [n129] In Plains Cotton Cooperative Ass'n v. Goodpasture Computer Service, [n130] a case factually similar to Whelan, the Fifth Circuit denied copyright protection to the structure of a cotton marketing program. [n131] The court noted that many of the similarities in the sequence and organization of the two programs were dictated by the cotton market for which they were designed. [n132] Commentators [*111] have likewise criticized Whelan. For example, Professor Melville B. Nimmer reasoned that "the crucial flaw in ... [the Whelan test's reasoning] is that it assumes that only one "idea,' in copyright law terms, underlies any computer program, and that once a separable idea can be identified, everything else must be expression." [n133]

III. Establishing Copyright Infringement

Copyright infringement requires proof of the plaintiff's ownership as well as proof that the defendant actually copied the plaintiff's work. [n134] Direct evidence of copying is usually nonexistent. In situations where there is less than actual "word-for-word" copying, a plaintiff can prove circumstantially that a defendant infringed by demonstrating proof of access to the plaintiff's work [n135] and substantial similarity between the works in question. [n136]

A. Proof of Substantial Similarity Does Not Require Proof that Most or All Parts of the Two Works Are Similar.

In order to compare and determine similarities, one must have [*112] at least a fundamental understanding of the programs' design, structure, and operation. Yet many lawyers and judges lack even a rudimentary technological background and will candidly confess their ignorance. As one court has observed, "the challenge to counsel to make comprehensible for the court the esoterica of bytes and modules is daunting." [n137] A structured method is therefore necessary. Legislation would, of course, be the ideal method for procuring such structure. However, since it has become clear that sufficient legislation is not forthcoming from Congress, it has become the responsibility of the courts to create and develop the law.

Courts face a variety of difficulties in determining whether computer programs are substantially similar. The Seventh Circuit, for instance, has characterized substantial similarity as necessarily subjective. [n138] The standard articulated by Learned Hand in Nichols v. Universal Pictures Corp. [n139] is the one most frequently employed to determine substantial similarity:

> Upon any work ... a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out ... There is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas," to which, apart from their expression, his property is never extended. ... As respects plays, the controversy chiefly centers upon the characters and sequence of incident, these being the substance. n140

Nichols concerned two plays which dealt with prejudice toward marriages between those of Jewish and Catholic faiths. n141 The court held that the defendant did not infringe the plaintiff's copyright with its play. Although both plays used the same ideas, the expressions were sufficiently different, in Judge Hand's view, to preclude a finding of infringement. n142 [*113]

B. Methods of Proving Substantial Similarity.

Substantial similarity can exist between works in two ways: "comprehensive nonliteral similarity" and "fragmented literal similarity." n143 Comprehensive nonliteral similarity is "a similarity not just as to a particular line or paragraph or other minor segment, but [where] the fundamental essence or structure of one work is duplicated in another." n144 Fragmented literal similarity consists of sporadic "word-for-word" similarity. n145

Courts have used three tests to determine comprehensive nonliteral similarity: (1) the ordinary observer test; (2) the extrinsic/intrinsic test; and (3) the total concept and feel test. The iterative test is the only test used to determine fragmented literal similarity.

1. The Ordinary Observer Test.

Regarding the ordinary observer test, n146 the First Circuit has remarked: "Specifically, the test is whether the accused work is so similar to the plaintiff's work that an ordinary reasonable person would conclude that the defendant unlawfully appropriated the plaintiff's protectable expression by taking material of substance and value." n147 Two works are substantially similar if "the ordinary [*114] observer, unless he [or she] sets out to detect the disparities, would be disposed to overlook them, and regard their aesthetic appeal as the same." n148 As a result, infringement will be found despite slight differences in the works. n149 "The sine qua non of the ordinary observer test ... is the overall similarities rather than the minute differences between the two works." n150

The Seventh Circuit applied the ordinary observer test in Atari, Inc. v. North American Philips Consumer Electronics Corp. n151 There, the court found that the defendant's K.C. Munchkin video game was substantially similar to the plaintiff's PAC-MAN game. n152 The court explained:

North American not only adopted the same basic characters but also portrayed them in a manner which made K.C. Munchkin appear substantially similar to PAC-MAN. The K.C. Munchkin gobbler has several blatantly similar features, including the relative size and shape of the "body," the V-shaped "mouth," its distinctive gobbling action (with appropriate sounds), and especially the way in which it disappears upon being captured. An examination of the K.C. Munchkin ghost monsters reveals even more significant visual similarities. In size, shape, and manner of movement, they are virtually identical to their PAC-MAN counterparts... Both games, moreover, express the role reversal and "regeneration" process with such great similarity that an ordinary observer could conclude only that North American copied plaintiff's PAC-MAN. [n153]

The Third Circuit criticized the ordinary observer test in Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc. [n154] The court maintained that this test was developed in cases dealing with literary and artistic works and involving no expert testimony. [n155] However, the test "is of doubtful value in cases involving computer programs on account of the programs' complexity and unfamiliarity [*115] to most members of the public." [n156] After rejecting the ordinary observer test, Judge Becker proceeded to adopt a variation of the iterative approach articulated in E.F. Johnson v. Uniden Corp. of America. [n157]

2. The Extrinsic/Intrinsic Test.

A second test for determining substantial similarity is the two part test employed by the Second Circuit in Arnstein v. Porter. [n158] The test has also been cited recently by the Ninth Circuit in Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp. [n159] The first part requires the fact-finder to discern whether there is sufficient similarity between the ideas expressed to infer use of the plaintiff's work by the defendant. [n160] This extrinsic prong depends "on specific criteria which can be listed and analyzed." [n161] Moreover, expert testimony is permitted for analytic purposes. [n162]

If the fact-finder finds that the ideas are sufficiently similar, then substantial similarity of the expressions must be determined; the "ordinary reasonable person" standard is used when making a determination concerning similarity. [n163] This intrinsic prong requires the fact-finder to determine substantial similarity without the assistance of any expert testimony. [n164]

The extrinsic/intrinsic test was also applied in Broderbund Software, Inc. v. Unison World, Inc. [n165] There, Judge Orrick concluded that the overall structure, sequence, and organization of screen displays in the defendant's "Printmaster" program were [*116] substantially similar to the plaintiff's "Print Shop" program. [n166] The controversy arose when Unison contacted Broderbund, which had acquired the rights in "Print Shop" (operable only on Apple computers), in the hope of developing an identical IBM version. [n167] Unison began working on the project and lawfully copied the Apple version of the program. [n168] However, the licensing negotiations broke down in the middle of the project. [n169] Unison nevertheless continued its development, and eventually marketed "Printmaster," which incorporated both the "Print Shop" interface as well as generous portions of the screen display. [n170] Although the court applied the extrinsic/intrinsic test, it is certainly arguable that, given the facts in Broderbund, the result would have been the same regardless of the standard employed. [n171]

The Third Circuit also criticized the extrinsic/intrinsic test in Whelan. [n172] Judge Becker found the test inappropriate when determining substantial similarity between computer programs. [n173] The court reasoned:

> The distinction between the two parts of the Arnstein test may be of doubtful value when the finder of fact is the same person for each step: that person has been exposed to expert evidence in the first step, yet she or he is supposed to ignore or "forget" that evidence in analyzing the problem under the second step. Especially in complex cases, we doubt that the "forgetting" can be effective when the expert testimony is essential to even the most fundamental understanding of the objects in question. [n174]

3. Total Concept and Feel Test.

The "total concept and feel" test originated in Roth Greeting Cards v. United Card Co. [n175] This test appears to be a hybrid of [*117] the ordinary observer and extrinsic/intrinsic tests. In Roth, the Ninth Circuit determined that the defendant had infringed the plaintiff's copyrighted greeting cards because the character, mood, artwork, message, and arrangement of the cards were substantially similar. [n176] The court observed:

> It appears to us that in total concept and feel the cards of [defendant] are the same as the copyrighted cards of [plaintiff] ... The remarkable similarity between Roth and United Cards in issue ... is apparent to even a casual observer. [n177] For example, one Roth card (Exhibit 9) has, on its front, a colored drawing of a cute moppet suppressing a smile and, on the inside, the words "i wuv you." With the exception of minor variations in color and style, defendant's card (Exhibit 10) is identical. Likewise, Roth's card entitled "I Miss you already" depicts a forlorn boy sitting on a curb weeping, with an inside message reading "... and You Haven't even Left ... " (Exhibit 7), is closely paralleled by United's card with the same caption, showing a forlorn and weeping man, and with the identical inside message (Exhibit 8). [n178]

The Ninth Circuit again applied the total concept and feel test in Sid & Marty Krofft Television Productions, Inc. v. McDonald Corp. [n179] The court ruled that McDonald's television commercials "captured the "total concept and feel' of the H. R. Pufnstuf television show." [n180] Writing for the court, Judge Carter continued:

> Even a dissection of the two works reveals their similarities. The "Living Island" locale of Pufnstuf and "McDonaldland" are both imaginary worlds inhabited by anthropomorphic plants and animals and other fanciful creatures. The dominant topographical features of the locales are the same: trees, caves, a

pond, a road, and a castle. Both works feature a forest with talking trees that have human faces and characteristics. The characters are also similar. Both lands are governed by mayors who have disproportionately large round heads dominated by long wide mouths. They are assisted by "Keystone Cop" characters. Both [*118] lands feature strikingly similar crazy scientists and a multi-armed creature. It seems clear that such similarities go beyond merely that of the idea into the area of expression. The use of the basic idea of the works does not inevitably result in such similarities. Certainly a jury applying an intrinsic test could find such similarities of expression substantial. n181

In addition, this test has been applied to computer programs in at least two instances. In both cases, courts found screen displays of the programs in question substantially similar. n182

Although the application of total concept and feel is applicable to computer screens, it may not be as helpful when comparing a program's structure, sequence, and organization. For example, one commentator has suggested:

> The ordinary observer inquiry was designed to determine general aesthetic similarities between the copyrighted work and the alleged copy ... Computer programs, however, have neither ordinary observers nor an easily perceived aesthetic appeal ... Since there are no ordinary observers of computer programs, the application of the ordinary observer standard to judge similarities in computer programs would be an entirely fictitious process. Compounding this problem is the absence of an easily perceived general "aura" or "feeling" that an ordinary observer could judge ... n183

This observation is well taken considering the fact that a computer program could be copied from another program, while looking nothing like the original to the untrained eye. n184 [*119]

Despite this difficulty, courts continue to apply this test. For example, in E.F. Johnson Co. v. Uniden Corp. of America, n185 the court reviewed evidence of similarity between the programs in question, and characterized one program's routines as "neat" and the other's as "rambling." n186 One commentator contends that "it appears that characteristics of a program's structure may in fact have an identifiable look and feel, at least to the eye of a trained programmer." n187 On the other hand, it is possible that a "rambling" program is merely a copy of a "neat" program which was simply made to look different. n188 [*120]

If the look and feel analysis ever achieves universal application, it may affect the manner in which programmers code their programs. As the Third Circuit explained in Whelan:

> Because efficiency is a prime concern in computer programs (an efficient program being obviously more valuable than a comparatively inefficient one), the arrangement of modules and subroutines is a critical factor ... The coding process is a comparatively small part of programming. By far the larger

portion of the expense and difficulty in creating computer programs is attributable to the development of the structure and logic of the program, and to debugging, documentation, and maintenance, rather than to the coding. n189

4. The Iterative Approach.

The three foregoing tests all require, to some extent, that programs be compared as a whole. The tests determine whether the infringing work "has gone beyond copying the "idea' [of plaintiff's work], and has borrowed its "expression.'" n190 It is possible, however, that a program may not infringe on another [*121] program under these three tests, yet still contain "word-for-word" copying. It is this "fragmented literal similarity" n191 that the iterative test seeks to prevent. At least one commentator has argued that, given the technical nature of computer programs, this standard is the proper test for determining substantial similarity in computer programs. n192

The iterative test was adopted in E.F. Johnson Co. v. Uniden Corp. of America. n193 E.F. Johnson Co. had developed a "Clearchannel LTR" logic trunk radio system n194 in 1980. n195 As Judge McLaughlin noted, "The heart of the ... LTR system is computer software contained in the mobile radios and repeaters." In 1985, Uniden introduced its model FTS 250T two-way 800 MHZ FM trunk mobile radio. n196 E.F. Johnson sought an injunction, arguing that the computer software in the defendant's system was identical to the software used by the plaintiff. n197 Regarding this, Judge McLaughlin observed:

> Because a copyrighted computer program is stored on a computer chip or disc well-hidden from public view, application of the ordinary observer test in a computer software context has proven problematical. The absence of an easily perceived general aura or feeling emanating from a silicon chip has led some commentators to suggest an "iterative" approach to substantial similarity ... Under the iterative approach, adopted in form if not name by several courts, the factfinder's focus shifts from the hypothetical ordinary observer's impressions of the "total concept and feel" of the copyrighted and allegedly infringed works to an analysis of the "quantitative and qualitative evidence of similarities" as gauged by the Court's evaluation of expert testimony. The fiction of the lay observer is thus abandoned in favor of an analysis of similarities and differences in the copyrighted and allegedly offending computer programs. n198

[*122]

According to the court, the iterative approach requires: (1) proof of copying, established either by actual copying or proof of access and sufficient similarity, and (2) proof that the infringing work is an exact duplication of substantial portions of the copyrighted work. n199

The Johnson court found several instances of duplication. First, thirty-eight of the forty-four subroutines were identical. n200 Second, both programs used 56-bit sampling, despite the fact that defendant's microprocessor could sample at a much higher rate. n201 Third, the sampling error table was copied, although this was not necessary for compatibility. [n202] Moreover, the H-matrix tables were identical, although this was also not necessary for compatibility. [n203] In addition, both programs contained three useless lines of code located in the same part of both programs. [n204] Finally, both programs contained the same error in the "select call prohibit." [n205] These similarities, in the court's view, constituted sufficient evidence of copyright infringement under the iterative test.

The Third Circuit employed the same approach in Williams Electronics, Inc. v. Artic International, Inc. [n206] There, the court found that "the extent of the copying could reasonably lead to no other conclusion" but that the defendant had copied the plaintiff's computer game, "DEFENDER." [n207] The court noted:

> There is overwhelming evidence in the present case that the Williams computer program has been copied in some form. The following facts, among others, manifest the similarities between the Williams program and that stored in the Artic [*123] memory devices:
>
> (1) The game created by the Artic circuit boards contains an error which was present in early versions of the Williams computer program - it displays the wrong score value for destroying a particular alien symbol;
>
> (2) The attract mode of both games displays a listing of high scores achieved by previous players alongside their initials, and Artic's game includes the initials of Williams employees, including its president, who initially achieved the highest scores on the DEFENDER game;
>
> (3) Using a laboratory developmental device, Williams' employees printed out a listing in code of the contents of the memory devices of both games. In excess of 85% of the listing is identical;
>
> (4) The Williams program provided that the words "Copyright 1980 - Williams Electronics" in code were to be stored in its memory devices, but were not to be displayed on the CRT at any time, thus providing a "buried" or hidden copyright notice. When the contents of Artic's memory devices were printed out by Williams' employees, the listings contained the "buried" Williams copyright notice in code. [n208]

This massive duplication ultimately led the court to its finding of infringement.

IV. An Analytical Method to Determine Substantial Similarity in Computer Programs

It is apparent that the long struggle among the courts to develop a workable rule has resulted in decisions that have "inevitably been ad hoc." [n209] The distinction between idea and expression nevertheless remains the critical element in an infringement action. Given the array of possible methods of analysis, there is a definite need for a systematic approach in order to avoid erroneous findings concerning infringement. Furthermore, such an approach would add a much needed measure of consistency to the law.

The goal of the analysis should be to separate the unprotectable [*124] abstract idea from the protectable expression. n210 Once the unprotectable aspects are singled out, they should no longer be considered. Eventually, after the filtering process is complete, a core of protectable material is left. This process makes the overwhelming task of comparing two complex computer programs more manageable and dispels the confusion which arises when unprotectable elements are similar, if not identical. This is particularly important in computer science, due to the notion that growth and development depend on free access to existing ideas, such as the structure or code of programs. n211

A. Why the Present Tests Are Inadequate.

To be sure, each of the tests described herein has its strengths. At the same time, however, each also has at least one fatal flaw which becomes particularly apparent when applied to computer programs. Each test is also quite vague and may result in either too much or too little copyright protection. The inadequacy of the present tests serves not only to circumvent the purposes of copyright law, but also to stifle the growth of the computer industry.

In Whelan Associates v. Jaslow Dental Laboratory, Inc., n212 the Third Circuit arguably reached the correct result on the basis of the evidence presented in the district court regarding the copying of protected expression. Despite this, the court created a broad and far sweeping rule which unnecessarily over-extended the scope of copyright protection. Moreover, the court failed to articulate a rule of law capable of guiding lower courts. The completely amorphous concept of "overall structure" is insufficient. n213 [*125]

In Whelan, there was no consideration whether the program in question was infringed under traditional copyright principles. The crucial flaw in Judge Becker's analysis is his assumption that only one idea underlies every computer program, and once that idea is found to be separable, everything else must be viewed as expression. In computer programs, especially multi-task software, it is not uncommon to find a plethora of ideas.

Although at least one court has adopted Whelan, n214 many have soundly rejected and criticized it. n215 The broad copyright protection afforded could potentially create a de facto monopoly of an idea by awarding a program excessive protection. Whelan may also afford greater protection to copyrights than patents, since unlike the case with patents, there is no inspection requirement for copyrights. n216 Such expansive protection is clearly contrary to the policies underlying copyright law. n217 Furthermore, such a rule [*126] offers no guidance to the computer industry. Such an ad hoc test would hamper development in the software field, because it would be unclear to what extent a program would be protectable. Consequently, a more predictable, bright line test is necessary.

The ordinary observer test, employed by the Seventh Circuit in Atari, Inc. v. North American Philips Consumer Electronics Corp., n218 should not be employed when analyzing the copyrightability of computer programs. First, it disallows expert testimony, and the determination of substantial similarity is completely left to the unguided trier-of-fact. Although the test may be useful for simple works, it is unsuited for works as analytically complex as computer programs. Second, the cases that have used the ordinary observer test have only compared screen displays. n219 This sends a clear message to the computer industry that it simply needs a different looking display. Also implicit in that assumption is that literal copying of source and object code is permissible.

A better test is the extrinsic/intrinsic test, first established in Arnstein v. Porter. n220 Although more predictable than the ordinary observer standard, the extrinsic/intrinsic test still has its flaws. While the first prong permits expert testimony in order to explain the similarity of ideas, the trier-of-fact is eventually left to determine what is protectable expression and whether that expression is substantially similar. Moreover, as the Third Circuit observed in Whelan, the two part extrinsic/intrinsic test "may be of doubtful value when the finder of fact is the same person for each [*127] step ...." n221 In addition, it only appears useful when analyzing screen displays. n222 It is unclear how the test would be applied to the analysis of source or object code. This is, of course, the same weakness implicit in the ordinary observer test.

The total concept and feel test, articulated by the Ninth Circuit in Roth Greeting Cards v. United Card Co., n223 is a hybrid of the ordinary observer and the extrinsic/intrinsic tests, and fails, not surprisingly, for the same reasons which rendered those tests inadequate. The words "total concept and feel" are vague and may over-extend copyright protection. However, as clarified recently in Brown Bag Software v. Symantec Corp., n224 total concept and feel is certainly a step in the right direction. The court in Brown Bag, by applying a version of the extrinsic/intrinsic test, permitted expert testimony to dissect and define the scope of copyright protection. n225 Nonetheless, Brown Bag still failed to eliminate the vague "look and feel" language. This language provides absolutely no guidance for lower courts and should not be retained. n226 While the look and feel test may be appropriate when analyzing greeting cards n227 or fantasy lands, n228 it proves inadequate when analyzing source and object code. This is even more apparent if the programs are written in different languages. In order to copy [*128] a program, a programmer would simply have to copy it into a different language. In this context, it is impossible to examine the similarities meaningfully without the assistance of expert testimony.

The iterative approach, as used in E.F. Johnson Co. v. Uniden Corp. of America, n229 seems to complicate, rather than simplify, the analysis. This two prong test has drawbacks similar to the other tests. The first part leaves the trier-of-fact to determine "use" with absolutely no guidance. It is less than clear why we should establish "use" as a criterion, since a person can use a program to get an idea without actually committing infringement. The second prong is even more ambiguous. Although literal similarity is necessary, there is no indication of how much infringement is actually necessary to constitute infringement. Every computer program that is compatible with the same computer system will naturally have some identical code. This is dictated by the system's operating system. n230 It remains unclear whether this literal copying would be considered when determining substantial similarity under the iterative approach.

B. The Abstractions Test Is Ideal to Determine Substantial Similarity of Computer Programs.

The most reliable test a court can utilize in separating unprotectable ideas and protectable expression within computer programs is the "abstractions test," n231 articulated by Learned Hand in Nichols v. Universal Pictures. n232 Judge Hand stated:

Upon any work, and especially upon a play, a great number of patterns of increasing generality will fit equally well, as more and more of the incident is left out. The last may perhaps be no more than the most general statement of what the play is [*129] about, and at times consist only of its title; but there is a point in this series of abstractions where they are no longer protected, since otherwise the playwright could prevent the use of his "ideas" to which, apart from their expression, his property is never extended. [n233]

Although slightly vague, the abstractions test is nevertheless the best approach available for analyzing computer programs. The Ninth Circuit has intimated that "no court or commentator ... has been able to improve upon Judge Learned Hand's famous "abstractions test'...." [n234]

The Second Circuit recently breathed new life into this test by applying it to computer programs in Computer Associates International, Inc. v. Altai, Inc. [n235] Computer Associates had marketed CA-SCHEDULER, a program designed to create a schedule specifying when the computer should run various tasks, and then to control the computer as it executes this schedule. [n236] Contained within CA-SCHEDULER was a wholly integrated compatibility component ("ADAPTER") which functioned by translating the language of a given program into the particular language that the computer's operating system could understand. [n237] By including such a component, the computer user was able to change or use multiple operating systems without changing software, thus saving both time and money. [n238]

In 1982, Altai began marketing ZEKE, its own scheduling program. [n239] Because of its success, Altai desired to rewrite ZEKE so that it could be run on other operating systems. [n240] An Altai employee, who had formally worked for Computer Associates, proceeded to rewrite ZEKE, using a copy of Adapter's source code which he had knowingly obtained in violation of his Computer Associates's employee agreement. [n241] Altai, his employer at the [*130] time, was unaware of his use of the computer program. [n242] The programmer eventually developed OSCAR 3.4, a compatibility component, which utilized thirty percent of OSCAR's code. [n243]

When Computer Associates discovered that Altai may have misappropriated parts of ADAPTER, it secured a copyright and instituted suit. [n244] Altai responded by rewriting Oscar in an attempt to purge it of those parts that had been copied from ADAPTER. [n245] The result was OSCAR 3.5. [n246] Altai subsequently sold OSCAR 3.5, and discontinued the sale of OSCAR 3.4. [n247] After a six day trial, a jury found Altai liable for infringing OSCAR 3.4. [n248] It did not, however, find Altai liable for infringing OSCAR 3.5. [n249] On appeal, after Altai had conceded infringement with respect to OSCAR 3.4, [n250] Computer Associates asserted that the district court had applied the wrong legal standard in erroneously finding that Altai had not infringed OSCAR 3.5. [n251] The Second Circuit upheld the district court's determination. [n252] The court utilized well-established copyright principles in devising a three part test for determining infringement:

In ascertaining substantial similarity under this approach, a court would first break down the allegedly infringed program into its constituent structural parts. Then, by examining each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all non-protectable material. Left with a kernel, or possible kernels, of creative expression after following this process of [*131] elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program. The result of this comparison will determine whether the protectable elements of the programs at issue are substantially similar so as to warrant a finding of infringement. [n253]

This abstractions test is readily adaptable to the analysis of computer software because most programmers adopt a "top-down" approach to programming. [n254] In utilizing this approach, the genesis of a particular program is its intended purpose. A specific outline is then created, based upon the end user's particular needs. Data structures and algorithms are developed for this outline. The entire logic of the program is then flow charted. [n255] This chart serves as a guide to develop subroutines. Finally, the source code, incorporating all the prior parts, is encoded. [n256] Both the limitations of the machines on which programs run as well as the formalities of a language contribute to a top-down approach.

This approach mirrors the approach used in the abstractions test. The initial general description is an unprotectable idea, while the final program is protectable expression. At some point during the process, the line between idea and expression is crossed. It is this line that the abstractions test attempts to draw.

C. Criticism of the Abstractions Test: Are Programs Science or Art?

To be sure, the application of this test to computer programs is debatable. Whether one views the abstractions test as applicable to computer programs depends ultimately on the observer's personal view of computer programs. How observers view the protection of computer programs hinges on whether they view computer programs as either functional science or art. As the Second Circuit remarked in Computer Associates: "the essentially utilitarian nature of a computer program further complicates the [*132] task of distilling its idea from its expression. In order to describe both computational processes and abstract ideas, its content "combines creative and technical expression'." [n257]

At least one commentator, for instance, has criticized Computer Associates, [n258] and it appears as though that commentator views programs as science rather than art. This school of thought advocates that there are not as many artistic choices in a computer program as compared to a literary work because of a variety of external factors, such as a particular programming language, and the unique requirements of a particular computer. In addition, there are many common programming techniques used throughout the industry. Moreover, the nature of the industry poses a particular dilemma. Programs are created to satisfy a particular need. Several manufacturers may be working on similar programs in an attempt to capitalize on this need. It is possible, therefore, that the programs independently created may nonetheless be strikingly similar because of the common need they are designed to meet.

One argument against the abstractions test that is consistent with this approach is that proving substantial similarity of computer programs is extremely difficult on account of a variety of technical difficulties. Because many programs are incomprehensible to a judge or jury, experts have to be retained. Analyzing substantial similarity is difficult, if not impossible, especially when allegations extend beyond literal copying. The trier-of-fact has to understand the design, structure, and function of both programs, and analyze accordingly.

A more plausible view is that programs should be seen as artistic expression rather than utilitarian science. There are two ways of viewing a particular program, the user's view and the [*133] programmer's view. [n259] One commentator has analogized computer programs to musical compositions. [n260] From the user's view, a program is nothing more than a means to get a computer to act in a certain way. [n261] The user's view is similar to that of a person listening to music. [n262] The actual score of the music is of little interest to the listener. [n263] It is simply a means of enabling musicians to play together and produce the piece that is eventually heard. [n264] It is the result that matters to the listener. [n265] Likewise, the user of a computer program is concerned primarily with the results of the program. [n266]

The other perspective is that of the programmer whose view is similar to that of a composer. [n267] Each "writer" composes lines made up of symbolic characters. [n268] Both have the end result in mind but require an expression to achieve that goal. [n269] Because computer programs contain the necessary expression, they should be seen as artistic works with an aesthetic appeal. [n270]

It could be argued, therefore, that the abstractions test provides for an analysis of computer programs which is more precise and consistent than that afforded traditional literary works. Literary works are not created in an orderly fashion, and it would be more difficult to break a literary work down into sub-levels. Computer programs, on the other hand, have natural divisions, and, thus, more readily discernible levels of abstraction. Of course, the test is still not easy to apply, [n271] but it does help courts to separate [*134] ideas from expression, and to eliminate unprotectable portions. Once separated, the protectable elements can be compared for substantial similarity purposes.

V. Resolving the Idea/Expression Dichotomy: The Determination of What is Protectable

Even after Computer Associates, it is still unclear what is to be classified as protectable and unprotectable. Because each case will ultimately turn on its own facts, this uncertainty will likely persist. However, if the analytical approach is followed, certain general elements should be seen as "not protectable." These elements are treated as such for two reasons - merger and the external restraints placed on the programmer.

A. Certain Expressions Have Become Limited, and Should Not Be Protected.

In order to avoid providing monopolies over certain ideas, the merger doctrine prohibits the protection of expression when the underlying idea can be expressed in only one fashion. [n272] The concept of merger is limited and has only been extended to a few situations. [n273] Merger issues arise in computer programs in unusual ways. In theory, many ways exist to encode a certain part [*135] of a program, but efficiency concerns can make one or two choices so compelling that they virtually eliminate all other forms of expression.

Courts have recognized the existence of merger in the computer context. [n274] For example, computer programs often need to search and sort algorithms, a time consuming endeavor. Of course, the most efficient method of searching and sorting is the most desirable one. Today, after many years of experience, programmers recognize that some methods of sorting, such as a bubble sort, are more efficient than others. [n275] Common sense dictates that the programmer use the most efficient method possible. [n276] Therefore, such methods should be denied protection under the merger doctrine. [*136]

B. Lack of Originality and Scenes a Faire: Program Elements Dictated By External Considerations.

It is axiomatic that protection is afforded for original elements only. n277 Nonetheless, courts have recognized that many elements of a work will be dictated by the work's theme or setting. These scenes a faire are incidents or plots that necessarily follow from a common theme or setting. n278 "Because it is virtually impossible to write about a particular historical era or fictional theme without employing certain "stock' or standard literary devices, courts have held that scenes a faire are not copyrightable as a matter of law." n279 Under traditional copyright, scenes a faire analysis is applied most frequently to factual or historical works. n280

Scenes a faire is applicable to computer programs since it is often impossible to write a program designed to perform particular functions in a specific computing environment without first [*137] employing standard techniques. n281 Therefore, a court should move a step beyond the abstractions test, and eliminate from consideration those elements which should be classified as non-original scenes a faire. These include hardware and software standards, manufacturer's design standards, target industry practices, industry programming practices, and public domain elements. Once eliminated, the remaining protectable expression is then analyzed for substantial similarity.

1. Elements That Should Be Eliminated.

The first elements to be eliminated are those which are dictated by the particular computer on which the program is run. If it is to run successfully, the program must be compatible with the computer. Compatibility requires that a program be tailored to meet the machine's design standards. This concept is analogous to scenes a faire in literary works. Thus, two programs may contain many similarities which are not the product of independent creativity, but are nonetheless necessary, since programs have standard instructions for gaining access to the computer's internal function. n282

For instance, the hardware on IBM personal computers which generates video displays is designed to accept instructions, but only in a precise format of characters and graphics. The addresses of video display control registers and video RAM in the IBM PC are both standardized and must be accessed in a particular fashion in order to generate a display. n283 Any program that is to run on an IBM PC must contain these specific instructions if it is to generate [*138] a display on the monitor screen. n284 Because all programs that are IBM compatible would contain this exact code, it should be eliminated from the substantial similarity analysis.

Just as the hardware dictates certain elements, the software environment in which the program is developed or operates may also govern elements of the program's design. To be compatible, a program must communicate with the computer's operating system. n285 Compatibility will dictate certain aspects, such as the way in which the program accesses data files on disk, and the way in which it is initially called up by the user. n286 Thus, as with hardware restraints, similarities occur due to software constraints which may not be attributable to copying. n287

Furthermore, the programming language used may also dictate how certain codes will look. For example, the types of data structures available to a programmer will depend on the language, as will the mechanics for controlling the flow of data and the ordering of subroutines in the program. n288 Therefore, elements that are governed by software standards should also be eliminated.

In addition to these restraints, manufacturers establish standards for third party programmers to follow when designing software to run on their machines. Such standards are usually non-essential to the error-free operation of the computer. For example, the nature of the user interface and the style of screen displays are sometimes standardized. Numerous other "standards" exist in the computer industry for everything from communications' protocols, to keystroke sequences, to file formats. [n289] Compliance with such standards is not mandatory, although these standards are established so that a manufacturer can assure users a familiar, consistent interface, regardless of who actually writes the program.

Most standards do not result from any formal agreement, but, [*139] rather, from the commercial success of a particular product. [n290] For example, two programs intended to permit sideways printing on spreadsheets produced by the popular Lotus 1-2-3 program must contain routines to read and interpret the Lotus 1-2-3 file format. The wholesale duplication of user interfaces may constitute infringement if such interfaces are protectable expressions of an idea rather than mere compliance with standards. [n291] It is necessary, therefore, to eliminate such user interfaces from a determination of substantial similarity.

Similarly, the older IBM personal computer family [n292] centered around certain "BIOS service routines." [n293] These routines [n294] provided all of the necessary control functions and operations thought necessary by IBM. [n295] Such service routines included, among other things, the operation of time/date clock, [n296] the performance of the print-screen operation, [n297] the system start-up [*140] routine, [n298] and video display. [n299] When new PC models were designed, they contained BIOS services which were fully compatible with the services in other members of the family. [n300] Since programmers utilize these BIOS services, [n301] many aesthetic elements of any software designed to run on IBM PC's will appear nearly identical. Similarities that result from compliance with manufacturers' standards should not be considered evidence of illegal copying. Even more important than particular software, hardware, or industry standards, however, is the business and technical requirements of the user. For example, programs used on the New York Stock Exchange must comply with the rules and practices of the Exchange. [n302] Hence, similarity is to be expected. [n303] However, such similarities should play no role in determining copyright infringement. [n304] Some courts may already be realizing this. For example, one court upheld a denial of a preliminary injunction based on a showing by the defendant that the similarity between the two cotton marketing programs at issue was "dictated [,in part,] by the externalities of the cotton market." [n305] [*141]

2. Analysis of Remaining Similarities and Possible Defenses.

Once the unprotectable material is filtered out, the resulting comparison by the fact-finder becomes a value judgment, which requires the fact finder to assess the importance of the material copied. It is possible to find that the actual amount of protectable material is so small as to be considered de minimis copying. [n306] On the other hand, a minute amount of copying may constitute infringement due to the material's importance to the program. [n307] Moreover, it is also possible that this identical code could be the result of an independent creation, in which case no infringement would occur. [n308] Finally, if such copying is determined to be a fair use, there would, of course, be no infringement. In order to determine fair use, the court should consider "the amount and substantiality of the portion used in relation to the copyrighted work as a whole." [n309]

VI. Conclusion

Given the difficulty courts have had in articulating a clear standard, it is foolish to assume that any particular method is the correct one. As with any legal rule, the application of the analytical method may prove difficult. Nonetheless, this method, unlike many of the others, is consistent with traditional copyright rules and principles. It would certainly assist both judges and [*142] advocates in analyzing what would otherwise be an incomprehensible computer program.

In the abstract, the method is quite simple: eliminate from the analysis that which is unprotectable, and compare only that which is protectable. Of course, each case will turn on its facts and should be analyzed accordingly. Occasionally, it may also be necessary to reinforce the analysis with other, more limited tests. Nevertheless, the analytical method remains the most promising candidate for guiding courts through what has otherwise been an almost impossible task.

Legal Topics:

For related research and practice materials, see the following legal topics:
Computer & Internet LawCopyright ProtectionGeneral OverviewCopyright LawCivil Infringement ActionsInfringement OnlineDeterminationsCopyright LawSubject MatterGeneral Overview

FOOTNOTES:

n1. Joseph Deken, The Electronic Cottage 1 (1981).

n2. In the United States alone, there are over 14,000 enterprises writing computer programs for commercial distribution. Anthony L. Clapes, Software, Copyright, and Competition: The "Look and Feel" of the Law 19 (1989). See Statistical Abstract of the United States 761 (1991). In 1981, Americans purchased 1.1 million personal computers at a total cost of over $ 3.1 billion. Id. However, by 1985, sales climbed to 6.75 million machines at a cost of $ 17.1 billion, and, by 1988, sales had reached 9.5 million at a cost of approximately $ 27.7 billion. Id. The percentage of American households with personal computers also rose during this time. Twenty-three percent of households had computers in 1989, as compared to 13% in 1985. See Trish Hall, Electronics: It's Not Home Without It, N.Y. Times, Mar. 29, 1990, at C6.

n3. See Howard Root, Note, Copyright Infringement of Computer Programs: A Modification of the Substantial Similarity Test, 68 Minn. L. Rev. 1264 (1984) (providing statistics).

n4. Id. at 1268-69. The author stresses that heightened computer protection laws are necessary to offset the dangerous increase in illegal copying which has accompanied the tremendous growth experienced by the computer software industry. Id. at 1264. Mr. Root's concerns resulted in his proposal of the "iterative" substantial similarity test. Id. at 1295. Under this standard, a plaintiff establishes a prima facie case of copyright infringement by offering evidence of access to and similarity between the plaintiff's work, and by demonstrating that the defendant's alleged copy is nothing more than a restatement of a substantial portion of the plaintiff's copyrighted program. Id.

n5. See Thomas L. Hazen, Contract Principles as a Guide for Protecting Intellectual Property Rights of Computer Software: The Limits of Copyright Protection, the Evolving Concept of Derivative Works, and the Proper Limits of Licensing Arrangements, 20 U.C. Davis L. Rev. 105, 105-06 (1986).

n6. See Michael C. Gemignani, the Law and the Computer 80 (1981); see also Martin T. Hillery, Note, The Second Circuit's Attempt to Define Copyright Protection for Computer Software: Is the Abstraction - Filtration - Comparison Test a Workable Solution?, 66 St. John's L. Rev. 1127, 1128-29 (1993). Hillery notes the seemingly endless confusion courts face as they attempt to distinguish between those elements which are unprotectable as "ideas" and those elements deserving of protection as "expression." Hillery, supra note 6, at 1129-30. Perhaps the great Learned Hand said it best when he remarked, "nobody has been able to fix that boundary, and nobody ever can." Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), cert. denied, 282 U.S. 902 (1931). There have, however, been attempts to establish this boundary. See, e.g., Final Report of the National Commission on New Technological Uses of Copyrighted Works 20 (1978)(the application of the "idea/expression" dichotomy to computers provides that "when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to an infringement.")[hereinafter Contu Final Report]. Concerning this, the Commission continued:

The use of specific language ... may be so essential to accomplish a desired result and so integrated with the use of a ... conception that the proper standard of infringement is one which will protect as far as possible the copyrighted language and yet allow the free use of the thought beneath the language.

Id. (quoting Continental Casualty Co. v. Beardsley, 253 F.2d 702, 706 (2d Cir. 1958)).

n7. See infra notes 134-208 and accompanying text.

n8. See infra notes 84-133 and accompanying text.

n9. See Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1236 n.28 (3d Cir. 1986) ("This test [to distinguish idea from expression] is necessarily difficult to state, and it may be difficult to understand in the abstract."), cert. denied, 479 U.S. 1031 (1987).

n10. See infra notes 146-57 and accompanying text.

n11. See infra notes 158-74 and accompanying text.

n12. See infra notes 175-89 and accompanying text.

n13. See infra notes 190-208 and accompanying text.

n14. Thorne D. Harris III, The Legal Guide to Computer Software Protection 20 (1985).

n15. Id. at 20-21.

n16. L.J. Kutten, Computer Software: Protection/Liability/Law/Forms 1.02, at 1-3 (1991).

n17. Id.

n18. Id.

n19. See Peter S. Menell, An Analysis of the Scope of Copyright Protection for Applications Programs, 41 Stan. L. Rev. 1045, 1051 (1989). Application programs are used primarily to complete data processing tasks. Id. Professor Menell notes that application programs include, among other things, bookkeeping, statistical and financial analysis, word processing, and video game programs. Id.

n20. Id.

n21. David Bender, 1 Computer Law: Evidence and Procedure 2.06[3], at 2-125 (1979).

n22. Note, Copyright Protection of Computer Program Object Code, 96 HARV. L. REV. 1723, 1724-25 (1983). High level languages are easier to understand because they "use words and mathematical symbols which are familiar to the programmer, such as "GO TO,' "DO,' "CONTINUE,' "ADD,' "+,' "-,' etc." Bender, supra note 21, 2.06[3], at 2-127.

n23. Copyright Protection of Computer Program Object Code, supra note 22, at 1725. For example, to compute a value Z constituting the sum of two other quantities denoted as X and Y in assembly would be:

```
L 2,X

A 2,Y

ST 2,Z
```

The same program in FORTRAN would be:

```
Z = X + Y
```

Bender, supra note 21, 2.06[3], at 2-130.

n24. Bender, supra note 21, 2.06[3], at 2-129 n.50. More skilled programmers can normally write a more efficient program in assembly, rather than high level, since the high level program will take more time to run. Id.

n25. See Copyright Protection of Computer Program Object Code, supra note 22, at 1724; cf. Bender, supra note 21, 2.06[3], at 2-129 ("most coding today is in higher level languages").

n26. Source code is copyrightable. See, e.g., Stern Elecs., Inc. v. Kaufman, 669 F.2d 852, 855 n.3 (2d Cir. 1982) ("Written computer programs are copyrightable as literary works.") (citing 1 Melville Nimmer, Nimmer on Copyright 2.04[C] (1981)); Apple Computer, Inc. v. Franklin Computer Corp., 545 F. Supp. 812, 822 n.15 (E.D.Pa. 1982) ("Programs written in source code are generally conceded to be copyrightable."), rev'd on other grounds, 714 F.2d 1240 (3d Cir. 1983).

n27. See Kutten, supra note 16, 1.06, at 1-12.

n28. Id.

n29. Object code is also copyrightable. See, e.g., Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870, 876-77 (3d Cir. 1982) (using the plain language of 17 U.S.C. 117 to reject defendant's attempt to distinguish between source and object versions of a computer program for purposes of copyright protection); GCA Corp. v. Chance, 217 U.S.P.Q. (BNA) 718, 720 (N.D. Cal. 1982) (refusing to treat object code as separate code as separate from the source code, since the former is the encryption of the latter and the two together comprise the copyrighted work); see also James Canfield, Note, The Copyrightability of Object Code, 59 Notre Dame L. Rev. 412, 422 (1984) (discussing how courts have generally found object codes to be copies of the source code and therefore subject to protection under copyright law). But see Data Cash Sys., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979) (holding that a ROM (Read Only Memory) was not a "copy" of the source code under both the common law and the Copyright Act of 1909).

n30. Clapes, supra note 2, at 49-51 (describing a computer as nothing but a massive collection of switches which can be turned on and off).

n31. The following is an example of a binary object code program designed to add three numbers:

01000101 00101100 00100100 01101111

BENDER, supra note 21, 2.06[3], at 2-126.

n32. The terms "literal" and "non-literal" are used by Nimmer. See Melville B. Nimmer & David Nimmer, 3 Nimmer on Copyright: A Treatise on the Law of Literary, Musical and Artistic Property, and the Protection of Ideas 13.03 [A], at 13-29, 13-45 (1994).

n33. Pamela Hobbs, Methods of Determining Substantial Similarity in Copyright Cases Involving Computer Programs, 67 U. Det. L. Rev. 393, 396 (1990).

n34. Id.

n35. See Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc. 797 F.2d 1222, 1239 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987). Concerning this, the Third Circuit observed:

Although the Code does not use the terms "sequence," "order," or "structure," it is clear from the definition of compilation and derivative works, and the protection afforded them, that Congress was aware of the fact that the sequencing and ordering of materials could be copyrighted, i.e., that the sequence and order could be parts of the expression, not the idea of a work.

Id.

n36. Manufacturers Technologies, Inc. v. Cams, Inc., 706 F. Supp. 984, 991 (D. Conn. 1989). Cf. Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1132 (N.D. Cal. 1986) (holding that user interface is an unprotectable non-literal element).

n37. Some argue that computer programs should not be protected by copyright. See Pamela Samuelson, CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form, 1984 Duke L.J. 663. The "academic" argument against copyrightability asserts that full copyright protection of computer programs does not further the purpose behind copyright law, that is, the dissemination of ideas. Clapes, supra note 2, at 142. Many commentators have criticized this argument on several grounds. They contend, for instance, that the purpose behind copyright laws is not to promote the dissemination of ideas, but instead, "to promote the Progress of Science and the Useful Arts." Id. at 145 (quoting U.S. Const. art. I, 8, cl. 8).

n38. U.S. Const. art. I, 8, cl. 8.

n39. See Act of May 31, 1790, ch. 15, 1 Stat. 124 (repealed 1831).

n40. See id.

n41. See Act of Apr. 29, 1802, ch. 36, 2 Stat. 171 (repealed 1831).

n42. See Act of Aug. 18, 1856, ch. 169, 11 Stat. 138, 139.

n43. See Act of Mar. 3, 1865, ch. 126, 1-2, 13 Stat. 540.

n44. See Act of July 8, 1870, ch. 180, 86, 16 Stat. 198, 212 (repealed 1909).

n45. Act of Mar. 4, 1909, ch. 320, 4, 35 Stat. 1075, 1076.

n46. Id., 5.

n47. See Act of Aug. 24, 1912, ch. 356, 5(l)-(m), 37 Stat. 488 (current version at 17 U.S.C. 102(a)(6) (1988)).

n48. See Act of Oct. 15, 1971, Pub. L. No. 92-140, 1(b), 85 Stat. 391 (current version at 17 U.S.C. 102(a)(7) (1988)).

n49. See Act of Oct. 19, 1976, Pub. L. No. 94-553, 90 Stat. 2541 (current version at 17 U.S.C. 101-810 (1988)).

n50. 17 U.S.C. 102(a) (1988).

n51. Id.

n52. See H.R. Rep. No. 1476, 94th Cong., 2d Sess. 52 (1976), reprinted in 1976 U.S.C.C.A.N. 5659, 5665 [hereinafter House Report].

n53. Id.

n54. Id., reprinted in 1976 U.S.C.C.A.N. at 5666; see also Jeffery R. Benson, Note, Copyright

Protection for Computer Screen Displays, 72 Minn. L. Rev. 1123, 1126 (1988) (noting that works qualify as "fixed" so long as they are embodied in a physical medium, such as a book, record, tape, or a computer disk).

n55. See 1 Nimmer & Nimmer, supra note 32, 2.18[B]-[D], at 2-197 to 2-207 (discussing the development of the idea/expression dichotomy). The authors note that 102(b) of the 1976 Act is the codification of two cases, Baker v. Selden, 101 U.S. 99 (1879) and Mazer v. Stein, 347 U.S. 201 (1954). Id. 2.18[C], at 2-204 n.30. In Baker, the plaintiff sought copyright protection for a bookkeeping system which required blank forms. Baker, 101 U.S. at 100. These forms consisted of ruled lines and headings especially designed for use with the plaintiff's bookkeeping system. Id. at 99. The defendant had published a book with forms which provided the same results as the plaintiff's bookkeeping system, while employing a different arrangement of columns and headings. Id. The plaintiff did not argue that the defendant's forms were substantially similar to the plaintiff's forms. Instead, the plaintiff claimed that the defendant's forms embodied a bookkeeping system which was similar to the plaintiff's own system. Id. The Court found that the bookkeeping system could not be copyrighted through a copyright in the forms. Id. at 106. Where the bookkeeping system cannot be used without the forms, the forms cannot be copyrighted, since copyright protection for the forms would provide a monopoly on the underlying idea - the bookkeeping system. Id. at 104-05.

In Mazer, the Court narrowly construed Baker to hold only that the copying of an idea without also copying the expression of the idea (since the defendant had made a different arrangement of columns and used different headings) does not constitute an infringement. See 1 Nimmer & Nimmer, supra note 32, 2.18[D], at 2-206 (citing Mazer, 347 U.S. at 201).

n56. 17 U.S.C. 102(b) (1988).

n57. Id.

n58. See Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 741-42 (9th Cir. 1971) (denying copyright protection where the idea of a bee pin was inseparable from its expression).

n59. Id.

n60. U.S. Const. art. I, 8, cl. 8.

n61. 101 U.S. 99 (1879).

n62. Id. at 99-100.

n63. Id. at 100.

n64. Id. at 101.

n65. Id.

n66. Id. at 103.

n67. Id. at 104.

n68. Educational Testing Servs. v. Katzman, 793 F.2d 533, 539 (3d Cir. 1986) (rejecting defendant's use of merger principles to attack plaintiff's copyright) (quoting Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983)).

n69. Toro Co. v. R & R Products Co., 787 F.2d 1208, 1212 (8th Cir. 1986).

n70. See 17 U.S.C. 102(a) (1988). Section 102(a) included the following categories within "works of authorship": " (1) literary works; (2) musical works, including any accompanying words; (3) dramatic works, including any accompanying music; (4) pantomimes and choreographic works; (5) pictorial, graphic, and sculptural works; (6) motion pictures and other audiovisual works; and (7) sound recordings". Congress amended 102(a) in 1990 to encompass "architectural works." See 17 U.S.C. 102(a) (Supp. V 1993).

n71. 17 U.S.C. 117 (1976), amended by 17 U.S.C. 117 (Supp. IV 1980). In calling for the repeal of 117, CONTU contended that "such action was necessary in order to prevent any question concerning the impropriety of program piracy and to assure that all works of authorship are treated comparably ...." Contu Final Report, supra note 6, at 13.

n72. See H.R. Rep. No. 1476, 94th Cong., 2d Sess. 116 (1976), reprinted in 1976 U.S.C.C.A.N. 5659, 5731.

n73. See House Report, supra note 52, reprinted in 1976 U.S.C.C.A.N. at 5667.

n74. Act of Dec. 31, 1974, Pub. L. No. 93-573, 201(a)-(c), 88 Stat. 1873, 1874 (1974).

n75. CONTU Final Report, supra note 6, at 4. The Commission's purpose was to

make recommendations on legislation or procedures concerning:

(1) the reproduction and use of copyrighted works of authorship:

(A) in conjunction with automatic systems capable of storing, processing, retrieving, and transferring information, and

(B) by various forms of machine reproduction, not including production by or at the request of instructors for use in face-to-face teaching activities; and

(2) the creation of new works by the application or intervention of such automatic systems of machine reproduction.

Id.

n76. Id.

n77. Id. at 11. Realizing that the cost of developing computer programs far outweighs the cost of their duplication, the Commission discussed four options designed to stimulate the dissemination of programs:

1. the creator may recover all of its costs plus a fair profit on the first sale of the work, thus leaving it unconcerned about the later publication of the work; or

2. the creator may spread its costs over multiple copies of the work with some form of protection against unauthorized duplication of the work; or

3. the creator's costs are borne by another, as, for example, when the government or a foundation offers prizes or awards; or

4. the creator is indifferent to the cost and donates the work to the public.

The consequence of the first possibility would be that the price of virtually any program would be so high that there would necessarily be a drastic reduction in the number of programs marketed. In this country, possibilities three and four occur, but rarely outside of academic and government-sponsored research. Computer programs are the product of great intellectual effort and their utility is unquestionable. The Commission is, therefore, satisfied that some form of protection is necessary to encourage the creation and broad distribution of computer programs in a competitive market. The Commission's conclusion is that the continued availability of copyright protection for computer programs is desirable.

Id.

n78. Id. at 19 (quoting House Report, supra note 52, 1976 U.S.C.C.A.N. at 5670) (emphasis added). The Commission observed that "Copyright, therefore, protects [a computer] program so long as it remains fixed in a tangible medium of expression but does not protect the electro-mechanical functioning of a machine." Id. at 20.

n79. CONTU Final Report, supra note 6, at 12.

n80. Id. The actual recommendation declared:

Notwithstanding the provisions of 106, [which grants the copyright owner the exclusive rights to reproduce the copyrighted work], it is not an infringement for the rightful possessor of a copy of a computer program to make or authorize the making of another copy or adaption of 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 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..

n81. See Act of Dec. 12, 1980, Pub. L. No. 96-517, 10, 94 Stat. 3015, 3028 (current version at 17 U.S.C. 101, 117 (1988)). Congress' adoption of the 1980 Amendments has been criticized. See, e.g., Samuelson, supra note 37, at 748-49 (arguing that Congress erroneously approved the amendments which protect programs in machine readable form, because such programs do not disclose their contents and are inherently utilitarian).

n82. See J. Clark Kelso & Alexandra Rebay, Problems of Interpretation Under the 1980 Computer Amendment, 23 Santa Clara L. Rev. 1001 (1983) (examining how economic and political considerations influence judicial interpretation of the Act); Richard H. Stern, Another Look at Copyright Protection of Software: Did the 1980 Act Do Anything for Object Code?, 3 Computer L.J. 1 (1981) (discussing the impact of the 1980 Amendments on object code protection).

n83. See, e.g., Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521 (9th Cir. 1984) (discussing the legislative history behind the extension of copyright protection to computer programs); Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1253-54 (3d Cir. 1983) (reasoning that a determination that computer operating systems are copyrightable is consistent with the 1980 amendments "which reflect Congress' receptivity to new technology and its desire to encourage, through the copyright laws, continued imagination and creativity in computer programming"); Williams Elecs., Inc. v. Artic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982) (providing copyright protection to the video game DEFENDER); Midway Mfg. Co. v. Artic Int'l, Inc., 547 F. Supp. 999 (N.D. Ill. 1982) (applying the 1976 Copyright Act), aff'd, 704 F.2d 1009 (7th Cir.), cert. denied, 464 U.S. 823 (1983); GCA Corp. v. Chance, 217 U.S.P.Q. (BNA) 718 (N.D. Cal. 1982) (holding that computer programs in object code format are copyrightable), aff'd, 628 F.2d 1038 (7th Cir. 1980); Tandy Corp. v. Personal Micro Int'l, 524 F. Supp. 171 (N.D. Cal. 1981) (examining the language of the Copyright Act to the copyrightability of computer programs). But see Data Cash Sys., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979), aff'd on other grounds, 628 F.2d 1038 (7th Cir. 1980).

n84. 714 F.2d 1240 (3d Cir. 1983).

n85. Id. at 1249.

n86. See Richard Raysman & Peter Brown, "Apple v. Franklin' - Critical Issues Resolved, N.Y. L.J., Sept. 13, 1983, at 1, 2 (recognizing other judicial decisions holding that object code is a valid subject for copyright protection); Robert P. Merges, Apple v. Franklin: An Essay on Technology and Judicial Competence, 2 Yale L. & Pol. Rev. 62 (1983) (analyzing the Apple court's reliance on the language and legislative history of the 1976 Copyright Act and the 1980 Amendments).

n87. Apple, 714 F.2d at 1247.

n88. Id. at 1249.

n89. Id. at 1250-54.

n90. See, e.g., Brown Bag Software v. Symantec Corp., 960 F.2d 1465 (9th Cir. 1992) (successfully applying an analytic dissection of a computer program in a test for substantial similarity), 113 S.Ct. 198 (1992); Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) (upholding district court's finding that the non-literal elements of a rewritten computer program were not substantially similar to the original program); Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992) (allowing the "fair use" of a copyright program in order to gain an understanding of that program's unprotected elements); Johnson Controls, Inc. v. Phoenix Control Sys., Inc., 886 F.2d 1173 (9th Cir. 1989) (including in the copyright of a computer program the non-literal elements found to involve expression rather than ideas); Data East USA, Inc. v. Epyx, Inc., 862 F.2d 204 (9th Cir. 1988) (reversing a district court's finding that two karate video games were substantially similar); Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256 (affirming a denial against former employees of an injunction aimed at preventing them from using employer information and research), reh'g denied, 813 F.2d 407 (5th Cir.), cert. denied, 484 U.S. 821 (1987); M. Kramer Mfg. Co. v. Andrews, 783 F.2d 421 (4th Cir. 1986) (accepting audio-visual components of a video game as copyrightable materials); Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78 (D. Mass. 1992) (upholding copyright protection for the menu tree of a spreadsheet program); Apple Computer, Inc. v. Microsoft Corp., 759 F. Supp. 1444 (N.D. Cal. 1991) (denying copyright protection for the visual displays of computer user interfaces); Apple Computer, Inc. v. Microsoft Corp., 779 F. Supp. 133 (N.D. Cal. 1991) (holding that an "unprotectable" element should be included in the determination of substantial similarity); Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (D. Mass. 1990) (accepting the on-screen, aesthetic elements of a computer program as copyrightable); Manufacturers Techs., Inc. v. Cams, Inc., 706 F. Supp. 984 (D. Conn. 1989) (allowing a developer's screen displays as copyrightable elements); Broderbund Software v. Unison World, Inc., 648 F. Supp. 1127 (N.D. Cal. 1986) (protecting the audio-visual displays of the computer program "Print Shop"); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485 (D. Minn. 1985) (finding that a software program for a dispatch radio system satisfied the substantial similarity test); Q-Co Indus. v. Hoffman, 625 F. Supp. 608 (S.D.N.Y. 1985) (leaving unprotected four prompting program modules in computer software since use of them constituted the use of an idea rather than use of any unique expression).

n91. 462 F. Supp. 1003 (N.D. Tex. 1978).

n92. Id. at 1013.

n93. Id. at 1004-06.

n94. The program in question was based on the Framed Structure Analysis Program - IBM FRAN. Id. at 1005. IBM had relinquished its rights in FRAN and allowed it to enter the public domain. Id.

n95. Id. at 1006.

n96. Id.

n97. Id. at 1007-08.

n98. Id. at 1014 (quoting Long v. Jordan, 29 F. Supp. 287, 288 (N.D. Cal. 1939)).

n99. 605 F. Supp. 816 (M.D. Tenn. 1985).

n100. Id. at 829.

n101. Id. at 818.

n102. Id. at 819-20.

n103. Id. at 830.

n104. Id.

n105. Id.

n106. Id.

n107. 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

n108. Id. at 1239-40.

n109. Id. at 1225.

n110. Id. at 1226.

n111. Id.

n112. Id.

n113. Id. at 1227.

n114. Id. at 1248.

n115. Id. at 1249 app.

n116. Id. at 1238.

n117. Id.

n118. Id.

n119. Id. at 1239.

n120. Id.

n121. Id.

n122. 462 F. Supp. 1003 (N.D. Tex. 1978) (holding that formats were ideas and therefore unprotectable). See supra notes 91-98 and accompanying text.

n123. Whelan, 797 F.2d at 1240.

n124. Id.

n125. Id.

n126. 775 F. Supp. 544 (E.D.N.Y. 1991), aff'd, 982 F.2d 693 (2d Cir. 1992).

n127. Id. at 558.

n128. Id. at 559.

n129. Id. at 558.

n130. 807 F.2d 1256 (5th Cir.), cert. denied, 484 U.S. 821 (1987).

n131. Id. at 1262.

n132. Id. at 1264. Regarding this, the court observed that "appellees' witness testified that their cotton marketing program was designed to present the same information as is contained on a cotton recap sheet, within the confines imposed by use of a computer." Id. at 1264 n.4.

n133. 3 Nimmer & Nimmer, supra note 32, 13.03[F], at 13-127. See also Jack E. Brown, The Current Status of Copyright Protection for Computer Software and Some Patent Protection Parallels, 6 Computer L. & Prac. 170 (1990) (commenting that the distinction between "idea" and "expression" is losing clarity in the pursuit of more conservative copyright decisions). But see Note, Copyright Protection of Screen Displays After Lotus Development Corporation v. Paperback Software International, 4 DePaul Bus. L.J. 485, 497 (1992) ("as with Synercom, Plains Cotton can be reconciled with Whelan").

n134. M. Kramer Mfg. Co. v. Andrews, 783 F.2d 421, 445 (4th Cir. 1986).

n135. Access is generally not an issue. Since it can be easily proved, it is, in fact, regularly conceded. Most programs are readily available, and techniques exist to ascertain a program's structure, thus enabling a potential infringer to reconstruct the program. See Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992) (accessing copyrighted portions to discern non-copyrighted material did not constitute copyright infringement); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485 (D. Minn. 1985) (see supra note 90).

n136. Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1134 (N.D. Cal. 1986).

n137. Q-Co Indus., Inc. v. Hoffman, 625 F. Supp. 608, 610 (S.D.N.Y. 1985).

n138. Atari, Inc. v. North Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 614 n.6 (7th Cir.), cert. denied, 459 U.S. 880 (1982).

n139. 45 F.2d 119 (2d Cir.), cert. denied, 282 U.S. 902 (1931).

n140. Id. at 121.

n141. Id. at 120-21.

n142. Id. at 121-22.

n143. 3 Nimmer & Nimmer, supra note 32, 13.03[A], at 13-30 to 13-31, cited in Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1234 n.26 (3rd Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

n144. Id.

n145. Id.

n146. The ordinary observer test presupposes that an unprotectable idea is separable from protectable expression. See, e.g., Peter Pan Fabrics v. Martin Weiner Corp., 274 F.2d 487, 489 (2d Cir. 1960) (holding that "there can be no copyright in the "ideas' disclosed but only in their expression"); Atari, Inc. v. North Am. Philips Consumer Elecs. Corp., 672 F.2d 607, 614-15 (7th Cir. 1982) (explaining that "copyright laws preclude appropriation of only those elements of the work that are protected by the copyright... "It is an axiom of copyright law that ... protection ... extends only to the particular expression of an idea and never to the idea itself'."), cert. denied, 459 U.S. 880 (1982). Although easy in the abstract, such a distinction is difficult, if not impossible, for lay observers, thrust into the position of considering computer programs, to make.

n147. Concrete Mach. Co. v. Classic Lawn Ornaments, Inc., 843 F.2d 600, 607 (1st Cir. 1988) (quoting Atari, 672 F.2d at 614).

n148. Id. (quoting Peter Pan, 274 F.2d at 489).

n149. Id. at 608 (citing Atari, 672 F.2d at 618).

n150. Id. (citing Peter Pan, 274 F.2d at 489).

n151. 672 F.2d 607 (7th Cir. 1982).

n152. Id. at 617-18.

n153. Id. at 618.

n154. 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

n155. Id. at 1232.

n156. Id.

n157. 623 F. Supp. 1485 (D. Minn. 1985).

n158. 154 F.2d 464 (2d Cir. 1946).

n159. 562 F.2d 1157 (9th Cir. 1977) (holding that defendant's "McDonaldland" TV commercials infringed plaintiff's "H.R. Pufnstuf" children's TV show).

n160. Id. at 1164.

n161. Id.

n162. Id. On the other hand, the ordinary observer test does not permit the use of expert testimony. See Whelan, 797 F.2d 1222, 1232 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

n163. Sid & Marty Krofft Television, 562 F.2d at 1164.

n164. Id.

n165. 648 F. Supp. 1127 (N.D. Cal. 1986).

n166. Id. at 1136-37. Both programs in question were used to create customized greeting cards, signs, banners, and posters. Id.

n167. Id. at 1130.

n168. Id. at 1130-31.

n169. Id. at 1131.

n170. Id.

n171. See Clapes, supra note 2, at 150.

n172. 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

n173. Id. at 1232-33.

n174. Id. at 1232-33.

n175. 429 F.2d 1106 (9th Cir. 1970).

n176. Id. at 1110.

n177. This appears to be analogous to the ordinary observer test. See supra notes 146-57 and accompanying text.

n178. Id. at 1110 (footnote added).

n179. 562 F.2d 1157 (9th Cir. 1977).

n180. Id. at 1167.

n181. Id. at 1167 n.9.

n182. See Digital Communications Assocs., Inc. v. Softklone Distrib. Corp., 659 F. Supp. 449, 465-66 (N.D. Ga. 1987) (placing status screens side-by-side clearly revealed their substantial similarity, since the upper portions of the screens were "virtually identical" with the only difference being the inscription of the different name of each program); Broderbund Software, Inc. v. Unison World, 648 F. Supp. 1127, 1137 (N.D. Cal. 1986) ("The ordinary observer could hardly avoid being struck by the eerie resemblance between the screens of the two programs.").

n183. Root, supra note 3, at 1288-96.

n184. Compare the following two programs. The first is a FORTRAN program that computes the area of a triangle. The second is a PASCAL program which calculates the radius of a circle. The two programs are virtually identical, yet they appear very different.

FORTRAN: [SEE FORMUAL IN ORIGINAL]

n185. 623 F. Supp. 1485 (D. Minn. 1985).

n186. Id. at 1496.

n187. Hobbs, supra note 33, at 406.

n188. The following two BASIC programs produce identical results. The second was copied from the first.

Neat program:

```
10 INPUT "Please enter your name" N$

20 INPUT "Please enter your age", A%

30 IF A% > 30 THEN GOTO 50

40 PRINT N$ ", youth has dreams. Preserve your ideals."

45 END

50 PRINT N$ ", age brings wisdom. You have experience."

60 END
```

Rambling program:

```
10 KEY OFF:CLS:PRINT"THE AGE REFLECTOR"

20 PRINT"Copyright (C) 1994 by Don McGahn"
```

```
30 PRINT
40 REM
50 PRINT"This program will give advice based on age."
60 PRINT"What is your name?"
70 INPUT R$
80 REM
90 GOTO 120
100 INPUT Z%
110 GOTO 140
120 PRINT "Please enter your age."
130 GOTO 100
140 IF Z% < 31 THEN GOTO 210
150 GOTO 190
160 WIDTH 40:FOR C=1 TO 25
170 IF J=2 THEN 1140
180 DATA PLUS,MORE,REPOS,REST,RUE,STREET,SALE,DIRTY
190 PRINT R$ ", age brings wisdom. You have experience."
200 GOTO 230
210 PRINT R$ ", youth has dreams. Preserve your ideals."
220 CLS
230 END
```

n189. Whelan, 797 F.2d at 1230-31.

n190. Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487 (2d Cir. 1960).

n191. See supra notes 143-45 and accompanying text.

n192. Root, supra note 3, at 1265.

n193. 623 F. Supp. 1485 (D. Minn. 1985).

n194. Id. at 1496-97.

n195. Id. at 1487.

n196. Id.

n197. Id.

n198. Id. at 1493.

n199. Id.

n200. Id. at 1496-97.

n201. Id. at 1494.

n202. Id.

n203. Id. at 1495.

n204. Id. at 1495-96. Similar problems can befall map producers. See, e.g., Rockford Map Publishers, Inc. v. Directory Serv. Co., 768 F. Supp. 144 (1985) (affirming a finding of copyright infringement against Directory Service which had copied the output of Rockford Map in producing a plat map of Ford County, Illinois).

n205. E.F. Johnson, 623 F. Supp. at 1496.

n206. 685 F.2d 870 (3d Cir. 1982).

n207. Id. at 876.

n208. Id. at 876 n.6.

n209. Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 704 (2d Cir. 1992) (quoting Peter Pan Fabrics, Inc. v. Martin Weiner Corp., 274 F.2d 487, 489 (2d Cir. 1960)).

n210. See J. Dianne Brinson, Copyright Software: Separating the Protected Expression From Unprotected Ideas, A Starting Point, 29 B.C. L. Rev. 803 (1988) (discussing how (1) the avoidance of confusing protected expression with program code; and (2) the avoidance of confusing a program's unprotected ideas with its function are the best methods available for successfully separating ideas from expressions in computer programs).

n211. Cf. Wright v. Warner Books, Inc., 953 F.2d 731, 742 (2d Cir. 1991) (observing that ideas are "as free as air").

n212. 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

n213. Id. at 1233.

n214. See Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1133 (N.D. Cal. 1986).

n215. Kepner-Tregoe, Inc. v. Leadership Software, Inc., 12 F.3d 527, 536 (5th Cir. 1994) (explaining why Whelan was rejected in Plains Cotton); Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 705-06 (2d Cir. 1992) (criticizing Whelan for having an "outdated appreciation of computer science"); Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256, 1262 (5th Cir. 1987) (declining to embrace the broad scope of protection afforded in Whelan since the record below was only partially developed and the similarities of the computer programs at issue were dictated by the "externalities of the cotton market"), cert. denied, 484 U.S. 821 (1987). But see Engineering Dynamics, Inc. v. Structural Software, Inc. 26 F.3d 1335, 1341 (5th Cir. 1994) (generously describing Whelan as simply one approach to addressing an area of the law - the copyrightability of the nonliteral elements of a computer program - which is currently in a "state of creative ferment").

n216. After an application is made for a patent, an inspection is required of the work by the staff employed by the Patent and Trademark Office. "Applications filed in the Patent and Trademark Office and accepted as complete applications are assigned for examination to the respective examining groups having the classes of inventions to which the applications relate ...." 37 C.F.R. 1.101(a)(1994). Unlike the Patent and Trademark Office, the Copyright Office simply accepts registrations. "The effective date of registration ... is the date on which the application, deposit, and $ 10 fee have all been received in the Copyright Office, ... provided[] the claim is later determined to be acceptable for registration by the Registrar of Copyrights ...." 37 C.F.R. 202.4 (1994).

n217. The Supreme Court has observed that "the economic philosophy behind the [copyright] clause ... is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare ...." Mazer v. Stein, 347 U.S. 201, 219 (1954); see Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156 (1975) (finding the "ultimate aim" of the copyright clause to be the stimulation of "artistic creativity" for the general public welfare). Moreover, the Court has made it clear that the accrual of benefits to the author as a result of his or her work is little more than a "secondary" consideration. See United States v. Paramount Pictures, Inc., 334 U.S. 131, 158 (1948)("The sole interest of the United States and the primary object in conferring the [copyright] monopoly lie in the benefits derived by the public from the labors of authors.") (quoting Fox Film Corp. v. Doyal, 286 U.S. 123, 127 (1932)).

n218. 672 F.2d 607 (7th Cir. 1982).

n219. See supra notes 151-57 and accompanying text.

n220. 154 F.2d 464 (2d Cir. 1946).

n221. Whelan, 797 F.2d at 1232. The court further explained that "that person has been exposed to expert evidence in the first step, yet she or he is supposed to ignore or "forget' that evidence in analyzing the problem under the second step." Id. at 1232-33.

n222. See supra notes 165-74 and accompanying text.

n223. 429 F.2d 1106 (9th Cir. 1970).

n224. 960 F.2d 1465 (9th Cir. 1992).

n225. See id. at 1475.

n226. See E.F. Johnson Co. v. Uniden Corp., 623 F. Supp. 1485, 1501 n.16 (D. Minn. 1985) (criticizing the bifurcated extrinsic/intrinsic test as "problematical in a computer software context" because attempting to "gauge the "aura' or "feel' of a computer program imbedded on a silicon chip is doomed ab initio").

n227. See, e.g., Roth Greeting Cards v. United Card Co., 429 F.2d 1106, 1110 (9th Cir. 1970) (reasoning that even a "casual observer" could see the remarkable similarity between the two companies' greeting cards).

n228. See, e.g., Sid & Marty Krofft Television Prods., Inc. v. McDonald's Corp., 562 F.2d 1157, 1166-67 (9th Cir. 1977) (applying the intrinsic test of substantial similarity and recognizing that children would identify the McDonald's fantasy characters with Krofft's characters since the former captured the "total concept and feel" of the latter's show).

n229. 623 F. Supp. 1485 (D. Minn. 1985).

n230. See McGraw-Hill, Inc. v. Worth Publishers, Inc., 335 F. Supp. 415, 421 (S.D.N.Y. 1971).

n231. See, e.g., Ronald S. Laurie, Comment: Use of a "Levels of Abstraction" Analysis For Computer Programs, 17 AIPLA Q.J. 232 (1989) (stressing that computer programs should be described at several levels of abstraction, from the maximum level of conceptual generality (i.e., the "purpose" of the program) to code implementation in order to provide the conceptual framework necessary for an analysis of similarities on a number of levels).

n232. 45 F.2d 119 (2d Cir. 1930), cert. denied, 282 U.S. 302 (1931).

n233. Id. at 121.

n234. Sid & Marty Krofft Television, 562 F.2d at 1163.

n235. 982 F.2d 693 (2d Cir. 1992).

n236. Id. at 698.

n237. Id.

n238. Id. at 699.

n239. Id.

[240] Id.

[241] Id. at 699-700.

[242] Id. at 700.

[243] Id.

[244] Id.

[245] Id.

[246] Id.

[247] Id.

[248] Id.

[249] See Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544, 561 (E.D.N.Y. 1991), aff'd, 23 U.S.P.Q.2d (BNA) 1241 (2d Cir. 1992).

[250] Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 701 (2d Cir. 1992).

[251] Id. at 701.

[252] Id. at 714.

[253] Id. at 706.

[254] A "top down" approach is a systematic method which begins with general ideas, and is developed into a particular program. Kenneth E. Kendall & Julie E. Kendall, Systems Analysis and Design 282-83 (1988).

[255] Id.

[256] Id.

[257] Computer Assocs., 982 F.2d at 704 (quoting Peter G. Spivack, Comment, Does Form Follow Function? The Idea/Expression Dichotomy In Copyright Protection of Computer Software, 35 U.C.L.A. L. Rev. 723, 755 (1988) (citations omitted)).

[258] See, e.g., Hillery, supra note 6, at 1142 (arguing that while the Computer Associates court "attempted to provide a "pragmatic' analysis, it failed to establish a truly useful approach, and instead placed additional unnecessary burdens on the lower courts").

[259] See Clapes, supra note 2, at 43-44 (reasoning that a musical composition is similar to a computer program because a typical listener cannot read music and a typical computer programmer cannot read

computer code - both simply seek an aesthetically pleasing or useful result).

n260. See id.

n261. Id. at 43.

n262. Id. at 43-44.

n263. Id. at 44.

n264. Id.

n265. Id.

n266. Id.

n267. Id.

n268. Id.

n269. Id.

n270. Id.

n271. See Lotus, 788 F. Supp. at 91.

n272. See, e.g., Landsberg v. Scrabble Crossword Game Players, Inc., 736 F.2d 485, 489 (9th Cir.) (finding the similarity between the game strategy books in question to be "no more than the similarity that must unavoidably be produced by anyone who wishes to use and restate the unprotectable ideas contained in Landsberg's work."), cert. denied, 469 U.S. 1037 (1984); Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971) (holding that life-like bee pin jewelry is unprotectable because its idea is inseparable from its expression); see also 3 Nimmer & Nimmer, supra note 32, at 13.03[B], at 13-75 n.163. The authors observed:

> Given the dilemma either of protecting original expression even when that protection can be leveraged to grant an effective monopoly over the idea thus expressed, or of making the idea free to all with the concomitant result that the plaintiff loses all effective copyright protection even over the original expression used, copyright law chooses the latter course.

3 Nimmer & Nimmer, supra note 32, at 13-75.

n273. See Morrissey v. Procter & Gamble Co., 379 F.2d 675, 678 (1st Cir. 1967).

n274. See, e.g., Engineering Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1347 (5th Cir.

1994) ("the remarkable similarities in some of these formats may be due to the inherent qualities of the ideas expressed (merger)."); Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 709 (2d Cir. 1992) (finding support for applying the merger doctrine in cases which have addressed the issue of substantial similarity in the context of computer program structure); Johnson Controls, Inc. v. Phoenix Control Sys., Inc., 886 F.2d 1173, 1176 (9th Cir. 1989) ("Whether the non-literal components of a program, including the structure, sequence and organization, and user-interface, are protected depends on whether, ... [given] the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself."); Lotus Dev. Corp. v. Paperback Software, Inc. 740 F. Supp. 37, 59 (D. Mass. 1992) ("If a particular expression is one of a quite limited number of the possible ways of expressing an idea, then ... the expression is not copyrightable."); Atari Games Corp. v. Nintendo of Am., Inc., 18 U.S.P.Q.2d (BNA) 1935 (N.D. Cal. 1991); Telemarketing Resources v. Symantec Corp., 12 U.S.P.Q.2d (BNA) 1991 (N.D. Cal. 1989) (discussing the applicability of the merger doctrine), aff'd in part and vacated in part sub nom., Brown Bag Software v. Symantec Corp., 960 F.2d 1465 (9th Cir.), cert. denied, 113 S. Ct. 198 (1992); Broderbund Software, Inc. v. Unison World, Inc., 648 F. Supp. 1127, 1131-33 (N.D. Cal. 1986) (discussing the idea-expression dichotomy with respect to computer software); E.F. Johnson Co. v. Uniden Corp. of Am., 623 F. Supp. 1485, 1500-02 (D. Minn. 1985) (explaining the idea-expression dichotomy).

n275. Richard L. Bernacchi et al., 1 Bernacchi on Computer Law, 3.12.2, at 3-104 (1992).

n276. After all, if speed were not important computers would not be used in the first place!

n277. See Alfred Bell & Co. v. Catalda Fine Arts, Inc., 191 F.2d 99, 102 (2d Cir. 1951) (interpreting "original" in the context of a copyrighted work as requiring "that the particular work "owes its origin' to the "author'.") (citing Burrow-Giles Lithographic Co. v. Sarony, 111 U.S. 53, 57-58 (1833)); 17 U.S.C. 102(a) (1988).

n278. See, e.g., Schwarz v. Universal Pictures Co., 85 F. Supp. 270, 278 (S.D. Cal. 1945) (""Courts have held repeatedly that such similarities and incidental details necessary to the environment or setting of an action are not the material of which copyrightable originality exists'.") (quoting Cain v. Universal Pictures Co., 47 F. Supp. 1013, 1017 (S.D. Cal. 1942)); Hoehling v. Universal City Studios, Inc., 618 F.2d 972, 979 (2d Cir. 1980) ("It is virtually impossible to write about a particular historical era or fictional theme without employing certain "stock' or standard literary devices."), cert. denied, 449 U.S. 841 (1980).

n279. Hoehling, 618 F.2d at 979 (citations omitted).

n280. See Landsberg v. Scrabble Crossword Game Players, Inc., 736 F.2d 485, 489 (9th Cir.) (discussing Hoehling), cert. denied, 469 U.S. 1037 (1984). Concerning this doctrine, the Ninth Circuit explained:

Under [this] doctrine, a second author does not infringe even if he [or she] reproduces verbatim the first author's expression, if that expression constitutes "stock scenes or scenes that flow necessarily from common unprotectable ideas' ... because to hold otherwise would give the first author a monopoly on the commonplace of ideas behind the scenes a faire.

Id. at 489 (quoting See v. Durang, 711 F.2d 141, 143 (9th Cir. 1983).

n281. See, e.g., Data East USA, Inc. v. Epyx, Inc., 862 F.2d 204, 208 (9th Cir. 1988) (noting that copyright protection cannot be given to "elements of an expression that necessarily follow from an idea"); Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256, 1262 (holding that the organization and configuration of information delivered to a computer were not protected by copyright since they were ideas rather than expression), reh'g denied, 813 F.2d 407 (5th Cir.), cert. denied, 484 U.S. 821 (1987); Q-Co Indus., Inc. v. Hoffman, 625 F. Supp. 608, 616 (S.D.N.Y. 1985) (analogizing the use of a particular module in a prompting program to the use of "wheels for a car").

n282. See Bernacchi et al., supra note 275, 3.12.2, at 3-106.

n283. See Peter Norton, Programmer's Guide To The IBM PC 81-95 (1985).

n284. See Bernacchi, supra note 275, 3.12.2, at 3-106.

n285. Id. at 3-107.

n286. Id. at 3-108.

n287. See, e.g., McGraw-Hill, Inc. v. Worth Publishers, Inc., 335 F. Supp. 415 (S.D.N.Y. 1971) (limiting the range of expression in economics textbook to technical terminology associated with the subject).

n288. See Bernacchi, supra note 275, 3.12.2, at 3-108.

n289. Id. at 3-109.

n290. See, e.g., Jim Seymour, Who Owns the Standards?, PC Magazine, May 26, 1987, at 174 (commenting that the market "chides" vendors who stray from the established industry standards).

n291. See, e.g., Johnson Controls, Inc. v. Phoenix Control Sys., Inc., 886 F.2d 1173, 1175 (9th Cir. 1989) (reasoning that if a particular element of the computer program "qualifies as an "expression' of an idea, rather than the idea itself[,]" it may be protected by copyright); Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 68 (D. Mass. 1990) (holding that the Lotus 1-2-3 interface goes beyond the expression of the idea to the original, distinctive elements of expression which are, of course, copyrightable); Manufacturers Technologies, Inc. v. CAMS, Inc., 706 F. Supp. 984, 994 (D. Conn. 1989) (holding copyrightable the expression created where the plaintiff's user interface displayed the expression of how to create a cost estimate).

n292. Included within this family are the original PC, the XT, the PCjr., the Portable PC, and the PC AT. Although these models were extremely popular in the mid-1980's, they all share many similarities with the personal computers of today.

n293. See Norton, supra note 285, at 17.

n294. The BIOS services are invoked by interrupts. Id. at 160. These interrupt instructions point to a particular location in the interrupt vector table, which is represented by a "Dec" number, a "Hex" number, and a "service (hex)" number. Id.

n295. Id.

n296. To read the current clock count requires a Dec of 26, a Hex of 1A, and a service of 0. Id. at 228.

n297. Sending the screen contents to the printer requires a Dec of 5, a Hex of 5, and the service is also not applicable. Id. at 226.

n298. To reboot the computer, a Dec of 25 and a Hex of 19 is required. Id. at 228.

n299. Setting the video mode requires a Dec of 16, a Hex of 10, and a service of 0. Id. at 226.

n300. Id. at 160.

n301. Peter Norton lists three compelling reasons for the use of BIOS services. First, "it encourages good programming practices and it avoids some of the kludgy tricks that have been the curse of many other computers." Id. at 17. Moreover, it increases the chances of "programs working on every member of the PC family." Id. Finally, "it gives IBM more flexibility in making improvements and additions to the line of PC computers." Id.

n302. 3 Nimmer & Nimmer, supra note 32, 13.03[F], at 13-138.

n303. Id. 13.03[A], at 13-138 to 13-139.

n304. See id. 13.03[B], at 13-69 (criticizing Williams v. Arndt, 626 F. Supp. 571 (D. Mass. 1985)). In Williams, the district court determined that the plaintiff's copyrighted manual, which contained a "step-by-step method for trading in various commodities[,]" was infringed by the defendant's computer program which embodied the same method. Williams, 626 F. Supp. at 573.

n305. Plains Cotton Coop. Ass'n v. Goodpasture Computer Serv., Inc., 807 F.2d 1256, 1262 & n.4 (5th Cir.), cert. denied, 484 U.S. 821 (1987).

n306. See Apple Computer, Inc. v. Microsoft Corp., 821 F. Supp. 616 (N.D. Cal. 1993) (finding no substantial similarity where only four out of 150 minor elements survived the filtering out of unprotectable expression), aff'd in part and rev'd in part, 35 F.3d 1345 (9th Cir. 1994), cert. denied, 1995 U.S. LEXIS 1535 (U.S. Feb. 21, 1995); 3 Nimmer & Nimmer, supra note 32, 13.03[A], at 13-29 ("It is clear that slight or trivial similarities are not substantial and are therefore non-infringing.").

n307. See, e.g., Atari Games Corp. v. Nintendo of America, Inc., 30 U.S.P.Q.2d (BNA) 1401 (N.D. Cal. 1993) (finding a copyright infringement where the only copying was that of a "signal stream" which was used to allow a "key" program to open a "lock" program in order to permit use of unlicensed videogames).

n308. See, e.g., Novelty Textile Mills, Inc. v. Joan Fabrics Corp., 558 F.2d 1090, 1092 n.2 (2d Cir. 1977) ("The defendant may introduce evidence of independent creation which would rebut the